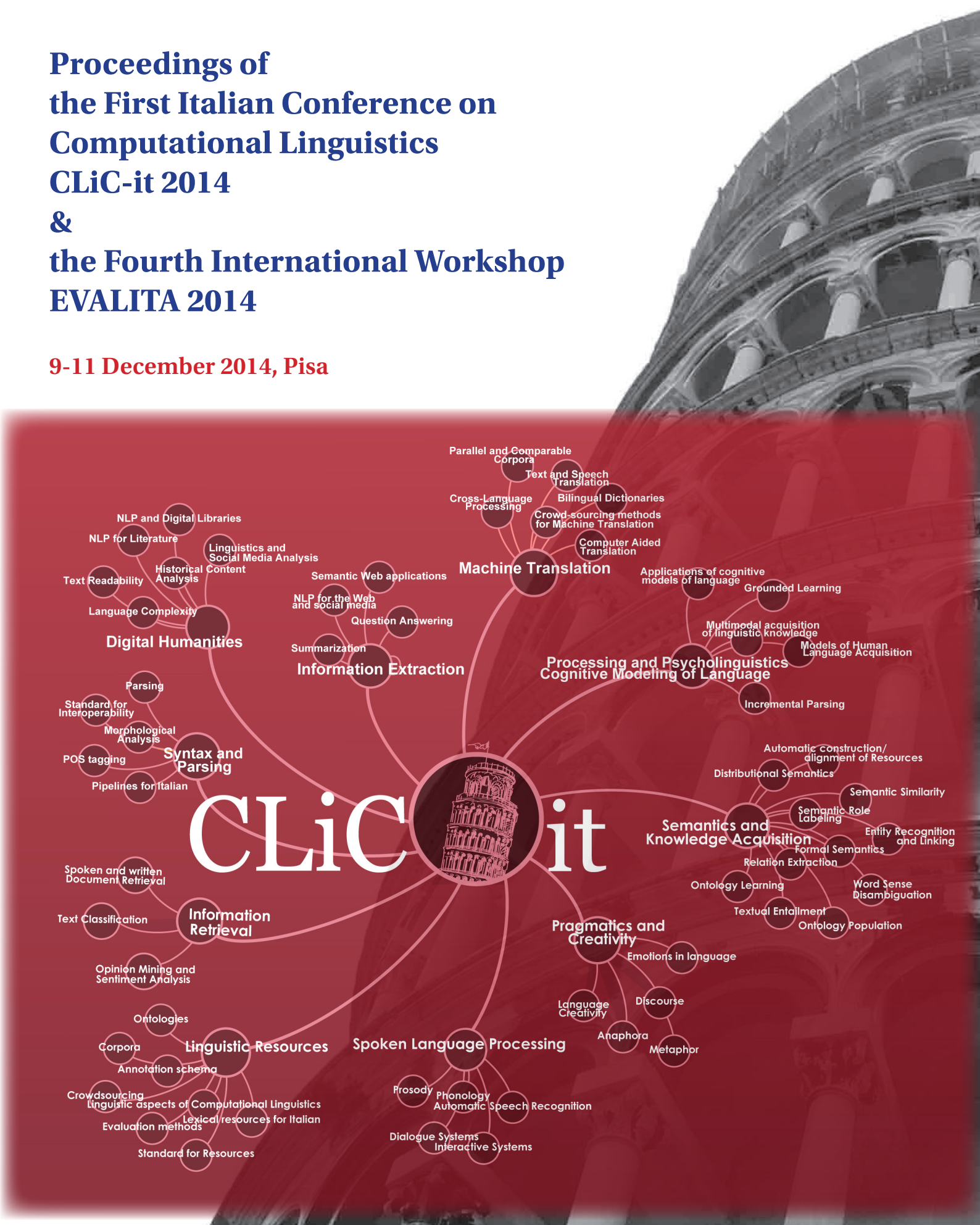
Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA 2014

9-11 December 2014, Pisa



PISA UNIVERSITY PRESS

Volume II

Fourth International Workshop EVALITA 2014

Proceedings

Editors
Cristina Bosco, Piero Cosi,
Felice Dell'Orletta, Mauro Falcone,
Simonetta Montemagni, Maria Simi

11th December 2014
Pisa, Italy

© Copyright 2014 by Pisa University Press srl
Società con socio unico Università di Pisa
Capitale Sociale Euro 20.000,00 i.v. - Partita IVA 02047370503
Sede legale: Lungarno Pacinotti 43/44 - 56126, Pisa
Tel. + 39 050 2212056 Fax + 39 050 2212945
e-mail: press@unipi.it
www.pisauniversitypress.it

ISBN 978-886741-472-7

Established in 2007, EVALITA (http://www.evalita.it) is the evaluation campaign of Natural Language Processing and Speech Technologies for the Italian language, organized around shared tasks focusing on the analysis of written and spoken language respectively. EVALITA's shared tasks are aimed at contributing to the development and dissemination of natural language resources and technologies by proposing a shared context for training and evaluation.

Following the success of previous editions, we organized EVALITA 2014, the fourth evaluation campaign with the aim of continuing to provide a forum for the comparison and evaluation of research outcomes as far as Italian is concerned from both academic institutions and industrial organizations. The event has been supported by the NLP Special Interest Group of the Italian Association for Artificial Intelligence (AI*IA) and by the Italian Association of Speech Science (AISV). The novelty of this year is that the final workshop of EVALITA is co-located with the 1st Italian Conference of Computational Linguistics (CLiC-it, http://clic.humnet.unipi.it/), a new event aiming to establish a reference forum for research on Computational Linguistics of the Italian community with contributions from a wide range of disciplines going from Computational Linguistics, Linguistics and Cognitive Science to Machine Learning, Computer Science, Knowledge Representation, Information Retrieval and Digital Humanities. The co-location with CLiC-it potentially widens the potential audience of EVALITA.

The final workshop, held in Pisa on the 11th December 2014 within the context of the XIII AI*IA Symposium on Artificial Intelligence (Pisa, 10-12 December 2014, http://aiia2014.di.unipi.it/), gathers the results of 8 tasks, 4 of which focusing on written language and 4 on speech technologies. In this EVALITA edition, we received 30 expressions of interest, 55 registrations and 43 actual submissions to 8 proposed tasks distributed as follows:

- Written language tasks: Dependency Parsing - DP (5), Evaluation of Events and Temporal Information - EVENTI (6), Sentiment Polarity Classification - SENTIPOLC (27), Word Sense Disambiguation and Lexical Substitution - WSD&LS (0);
- Speech tasks: Emotion Recogniton Task - ERT (2), Forced Alignment on Children Speech - FACS (1), Human and Machine Dialect Identification from Natural Speech and Artificial Stimuli - HMDI (0), Speech Activity Detection and Speaker Localization in Domestic Environments - SASLODOM (2).

23 participants (either as individual researchers or as academic institutions) submitted their results to one or more different tasks of the contest.

In this volume, the reports of the tasks' organizers and participants of EVALITA 2014 are collected.

As in previous editions, both the tasks and the final workshop were collectively organized by several researchers from the community working on Italian language resources and technologies. We thank all the people and institutions involved in the organization of the tasks, who contributed to the success of the event. A special thank is due to Francesco Cutugno (Università Degli Studi di Napoli Federico II) for his important contribution to the organization of the EVALITA Speech tasks. Thanks are also due to Manuela Sanguinetti (Università di Torino) for helping with the management of the EVALITA website, and to FBK for making the web platform available for this

edition as well. Last but not least, we thank our invited speaker, Ryan McDonald from Google, for agreeing to share his expertise on key topics of EVALITA 2014.

November 2014

EVALITA 2014 CO-CHAIRS

Cristina Bosco
Piero Cosi
Felice Dell'Orletta
Mauro Falcone
Simonetta Montemagni
Maria Simi

EVALITA 2014 Scientific coordination

- Cristina Bosco (Università di Torino)
- Felice Dell'Orletta (Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR, Pisa)
- Simonetta Montemagni (Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR, Pisa)
- Maria Simi (Università di Pisa)

EVALITA 2014 Scientific coordination for Speech Technology Evaluation

- Piero Cosi (Istituto di Scienze e Tecnologie della Cognizione - CNR, Padova)
- Mauro Falcone (Fondazione Ugo Bordoni)

EVALITA 2014 Steering Committee

Name	**Institution**	**Task**
Valerio Basile	University of Groningen, Netherlands	SENTIPOLC
Andrea Bolioli	CELI, Torino, Italy	SENTIPOLC
Cristina Bosco	Università di Torino, Italy	DP
Alessio Brutti	Fondazione Bruno Kessler, Trento, Italy	SASLODOM
Tommaso Caselli	VU Amsterdam, Netherlands	EVENTI
Piero Cosi	Istituto di Scienze e Tecnologie della Cognizione - CNR, Italy	FACS
Francesco Cutugno	Università di Napoli "Federico II", Italy	FACS
Felice Dell'Orletta	Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR, Pisa	DP
Vincenzo Galatà	Istituto di Scienze e Tecnologie della Cognizione - CNR, Italy	ERT, FACS
Monica Monachini	Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR, Pisa	EVENTI
Simonetta Montemagni	Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR, Pisa	DP
Malvina Nissim	University of Groningen and Università di Bologna, Netherlands, Italy	SENTIPOLC
Maurizio Omologo	Fondazione Bruno Kessler, Trento, Italy	SASLODOM
Antonio Origlia	Università di Napoli "Federico II", Italy	ERT, FACS
Viviana Patti	Università di Torino, Italy	SENTIPOLC
Mirco Ravanelli	Fondazione Bruno Kessler, Trento, Italy	SASLODOM
Antonio Romano	Università di Torino, Italy	HDMI
Paolo Rosso	Universitat Politècnica de València, Spain	SENTIPOLC
Claudio Russo	Università di Torino, Italy	HDMI
Manuela Sanguinetti	Università di Torino, Italy	DP
Maria Simi	Università di Pisa, Italy	DP
Manuela Speranza	Fondazione Bruno Kessler, Trento, Italy	EVENTI
Rachele Sprugnoli	Fondazione Bruno Kessler and University of Trento, Trento, Italy	EVENTI

Indice

WRITTEN LANGUAGE TASKS

Dependency Parsing

The Evalita 2014 Dependency Parsing task
Cristina Bosco, Felice Dell'Orletta, Simonetta Montemagni, Manuela Sanguinetti, Maria Simi 1

Dependency Parsing Techniques for Information Extraction
Giuseppe Attardi, Maria Simi 9

Comparing State-of-the-art Dependency Parsers for the EVALITA 2014 Dependency Parsing Task
Alberto Lavelli 15

Testing parsing improvements with combination and translation in Evalita 2014
Alessandro Mazzei 21

Evaluation of Events and Temporal Information

EVENTI. EValuation of Events and Temporal INformation at Evalita 2014
Tommaso Caselli, Rachele Sprugnoli, Manuela Speranza and Monica Monachini 27

Experiments in Identification of Italian Temporal Expressions
Giuseppe Attardi and Luca Baronti 35

HeidelTime at EVENTI: Tuning Italian Resources and Addressing TimeML's Empty Tags
Giulio Manfredi, Jannik Strötgen, Julian Zell and Michael Gertz 39

FBK-HLT-time: a complete Italian Temporal Processing system for EVENTI-Evalita 2014
Paramita Mirza and Anne-Lyse Minard 44

Sentiment Polarity Classification

Overview of the Evalita 2014 SENTIment POLarity Classification Task
Valerio Basile, Andrea Bolioli, Viviana Patti, Paolo Rosso and Malvina Nissim 50

UNIBA at EVALITA 2014-SENTIPOLC Task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features
Pierpaolo Basile and Nicole Novielli 58

ITGETARUNS A Linguistic Rule-Based System for Pragmatic Text Processing
Rodolfo Delmonte 64

Subjectivity, Polarity And Irony Detection: A Multi-Layer Approach
Elisabetta Fersini, Enza Messina, Federico Alberto Pozzi 70

IRADABE: Adapting English Lexicons to the Italian Sentiment Polarity Classification task
Irazú Hernández Farias, Davide Buscaldi and Belém Priego Sánchez 75

Linguistically–motivated and Lexicon Features for Sentiment Analysis of Italian Tweets
Andrea Cimino, Stefano Cresci, Felice Dell'Orletta and Maurizio Tesconi 81

The CoLing Lab system for Sentiment Polarity Classification of tweets
Lucia C. Passaro, Gianluca E. Lebani, Laura Pollacci, Emmanuele Chersoni and Alessandro Lenci 87

The FICLIT+CS@UniBO System at the EVALITA 2014 Sentiment Polarity Classification Task
Pierluigi Di Gennaro, Arianna Rossi and Fabio Tamburini 93

A Multiple Kernel Approach for Twitter Sentiment Analysis in Italian
Giuseppe Castellucci, Danilo Croce, Diego De Cao and Roberto Basili 98

Relying on intrinsic word features to characterise subjectivity, polarity and irony of Tweets
Francesco Barbieri, Francesco Ronzano and Horacio Saggion 104

Self-Evaluating Workflow for Language-Independent Sentiment Analysis
Arseni Anisimovich 108

SPEECH TASKS

Emotion Recogniton Task (ERT)

EVALITA 2014: Emotion Recognition Task (ERT)
Antonio Origlia, Vincenzo Galatà 112

A Preliminary Application of Echo State Networks to Emotion Recognition
Claudio Gallicchio, Alessio Micheli 116

Emotion Recognition with a Kernel Quantum Classifier
Fabio Tamburini 120

Forced Alignment on Children Speech (FACS)

Forced Alignment on Children Speech
Piero Cosi, Francesco Cutugno, Vincenzo Galatà and Antonio Origlia 124

The SPPAS participation to Evalita 2014
Brigitte Bigi 127

Human and Machine Dialect Identification from Natural Speech and Artificial Stimuli (HDMI)

Human and Machine Language / Dialect Identification from Natural Speech and Artificial Stimuli: a Pilot Study with Italian Listeners
Antonio Romano and Claudio Russo 131

Speech Activity Detection and Speaker Localization in Domestic Environments (SASLODOM)

SASLODOM: Speech Activity detection and Speaker LOcalization in DOMestic environments
Alessio Brutti, Mirco Ravanelli, Maurizio Omologo 139

The L2F system for the EVALITA-2014 speech activity detection challenge in domestic environments
Alberto Abad, Miguel Matos, Hugo Meinedo, Ramon F. Astudillo, Isabel Trancoso 147

Neural Networks Based Methods for Voice Activity Detection in a Multi-room Domestic Environment
Giacomo Ferroni, Roberto Bonfigli, Emanuele Principi, Stefano Squartini, and Francesco Piazza 153

10.12871/clicit201421

The Evalita 2014 Dependency Parsing task

Cristina Bosco[1], Felice Dell'Orletta[2], Simonetta Montemagni[2], Manuela Sanguinetti[1], Maria Simi[3]
[1]Dipartimento di Informatica - Università di Torino, Torino (Italy)
[2]Istituto di Linguistica Computazionale "Antonio Zampolli" - CNR, Pisa (Italy)
[3]Dipartimento di Informatica - Università di Pisa, Pisa (Italy)
{bosco,msanguin@di.unito.it},
{felice.dellorletta,simonetta.montemagni@ilc.cnr.it}
simi@di.unipi.it

Abstract

English. The Parsing Task is among the "historical" tasks of Evalita, and in all editions its main objective has been to define and improve state-of-the-art technologies for parsing Italian. The 2014's edition of the shared task features several novelties that have mainly to do with the data set and the subtasks. The paper therefore focuses on these two strictly interrelated aspects and presents an overview of the participants systems and results.

Italiano. *Il "Parsing Task", tra i compiti storici di Evalita, in tutte le edizioni ha avuto lo scopo principale di definire ed estendere lo stato dell'arte per l'analisi sintattica automatica della lingua italiana. Nell'edizione del 2014 della campagna di valutazione esso si caratterizza per alcune significative novità legate in particolare ai dati utilizzati per l'addestramento e alla sua organizzazione interna. L'articolo si focalizza pertanto su questi due aspetti strettamente interrelati e presenta una panoramica dei sistemi che hanno partecipato e dei risultati raggiunti.*

1 Introduction

The Parsing Task is among the "historical" tasks of Evalita, and in all editions its main objective has been to define and improve state-of-the-art technologies for parsing Italian (Bosco and Mazzei, 2013). The 2014's edition of the contest features two main novelties that mainly deal with the internal organization into subtasks and the used data sets.

From Evalita 2007 onwards, different subtasks have been organized focusing on different aspects of syntactic parsing. In Evalita 2007, 2009 and 2011, the tracks were devoted to dependency parsing and constituency parsing respectively, both carried out on the same progressively larger dataset extracted from the Turin University Treebank (TUT[1]), which was released in two formats: the CoNLL–compliant format using the TUT native dependency tagset for dependency parsing, and the Penn Treebank style format of TUT–Penn for constituency parsing. This allowed the comparison of results obtained following the two main existing syntactic representation paradigms as far as Italian is concerned.

In order to investigate the behaviour of parsing systems trained on different treebanks within the same representation paradigm, in 2009 the dependency parsing track was further articulated into two subtasks differing at the level of used treebanks: TUT was used as the development set in the main subtask, and ISST–TANL (originating from the ISST corpus, (Montemagni et al., 2003)) represented the development set for the pilot subtask. Comparison of results helped to shed light on the impact of different training resources, differing in size, corpus composition and adopted annotation schemes, on the performance of parsers.

In Evalita 2014, the parsing task includes two subtasks focusing on dependency parsing only, with a specific view to applicative and multilingual scenarios. The first, henceforth referred to as *Dependency Parsing for Information Extraction* or DPIE, is a basic subtask focusing on standard dependency parsing of Italian texts, with a dual evaluation track aimed at testing both the performance of parsing systems and their suitability to Information Extraction tasks. The second subtask, i.e. *Cross–Language dependency Parsing* or CLaP, is a pilot multilingual task where a source Italian treebank is used to train a parsing model which is then used to parse other (not necessarily typologically related) languages.

[1]http://www.di.unito.it/~tutreeb

Both subtasks are in line with current trends in the area of dependency parsing. In recent years, research is moving from the analysis of grammatical structure to sentence semantics, as testified e.g. by the SemEval 2014 task "Broad-Coverage Semantic Dependency Parsing" aimed at recovering sentence–internal predicate–argument relationships for all content words (Oepen et al., 2014): in DPIE, the evaluation of the suitability of the output of participant systems to information extraction tasks can be seen as a first step in the direction of targeting semantically–oriented representations. From a multilingual perspective, cross–lingual dependency parsing can be seen as a way to overcome the unavailability of training resources in the case of under–resourced languages. CLaP belongs to this line of research, with focus on Italian which is used as source training language.

As far as the data set is concerned, in Evalita 2014 the availability of the newly developed *Italian Stanford Dependency Treebank* (ISDT) (Bosco et al., 2013) made it possible to organize a dependency parsing task with three main novelties with respect to previous editions:

1. the annotation scheme, which is compliant to *de facto* standards at the level of both representation format (CoNLL) and adopted tagset (Stanford Dependency scheme, (de Marneffe and Manning, 2008));

2. its being defined with a specific view to supporting Information Extraction tasks, a feature inherited from the Stanford Dependency scheme;

3. the size of the data set, much bigger (around two times larger) than the resources used in previous Evalita campaigns.

The paper is organized as follows. The next section describes the resources that were used and developed for the task. In sections 3 and 4, we will present the subtasks, the participants' systems approaches together with achieved results.

2 A new dataset for the Evalita Parsing Task

Over the last few years, Stanford Dependencies (SD) have progressively gained the status of *de facto* standard for dependency–based treebank annotation (de Marneffe et al., 2006; de Marneffe and Manning, 2008). The *Italian Stanford Dependency Treebank* (ISDT) is the standard-compliant treebank for the Italian language (Bosco et al., 2013; Simi et al., 2014), which was built starting from the *Merged Italian Dependency Treebank* (MIDT) (Bosco et al., 2012), an existing dependency-based Italian treebank resulting in its turn from the harmonization and merging of smaller resources (i.e. TUT and ISST–TANL, already used in previous Evalita campaigns) adopting incompatible annotation schemes. ISDT originates as the result of a joint effort of three research groups based in Pisa (Dipartimento di Informatica – Università di Pisa, and Istituto di Linguistica Computazionale "Antonio Zampolli" – CNR) and in Torino (Dipartimento di Informatica – Università di Torino) aimed at constructing a larger and standard-compliant resource for the Italian language which was expected to create the prerequisites for crucial advancements in Italian NLP.

ISDT has been used in both DPIE and CLaP Evalita 2014 tasks, making it possible to compare parsers for Italian trained on a new, standard-compliant and larger resource, and to assess cross-lingual parsing results using a parser trained on an Italian resource.

The composition of the ISDT resource released for development in both tasks is as follows:

- a data set of around 97,500 tokens, obtained by conversion from TUT, representative of various text genres: legal texts from the Civil code, the Italian Constitution, and European directives; newspaper articles and wikipedia articles;

- a data set of around 81,000 tokens, obtained by conversion from ISST–TANL, including articles from various newspapers.

For what concerns the representation format, ISDT data comply with the standard CoNLL-X format, with UTF-8 encoding, as detailed below:

- sentences are separated by an empty line;

- each token in a sentence is described by ten tab–separated columns;

- columns 1–6 are provided by the organizers and contain: token id, word form, lemma, coarse-grained PoS, fine-grained PoS, and morphology;

- parser results are reported in columns 7 and 8 representing respectively the head token id and the dependency linking the token under description to its head;
- columns 9-10 are not used for the tasks and contain an underscore.

The used annotation scheme follows as close as possible the specifications provided in the SD manual for English (de Marneffe and Manning, 2008), with few variations aimed to account for syntactic peculiarities of the Italian language: the Italian localization of the Stanford Dependency scheme is described in detail in Bosco et al. (2013). The used tagset, which amounts to 41 dependency tags, together with Italian-specific annotation guidelines is reported in the dedicated webpage[2]. For what concerns the rendering of copular verbs, we preferred the standard option of making the copular verb the head of the sentence rather than the so-called Content Head (CH) option, that treats copular verbs as auxiliary modifiers of the adjective or predicative noun complement.

As stated in de Marneffe and Manning (2008), different variants of the typed dependency representation are available in the SD annotation scheme. Among them it is worth reporting here:

- the *basic* variant, corresponding to a regular dependency tree;
- the *collapsed* representation variant, where dependencies involving prepositions, conjunctions as well as information about the antecedent of relative pronouns are collapsed to get direct dependencies between content words. This collapsing is often useful in simplifying patterns in relation extraction applications;
- the *collapsed dependencies with propagation of conjunct dependencies* variant including – besides collapsing of dependencies – also the propagation of the dependencies involving conjuncts.

Note that in the collapsed and propagated variants not all words in a sentence are necessarily connected nor form a tree structure: this means that in these variants a sentence is represented as a set of binary relations (henceforth, we will refer to this representation format as RELS output). This is a semantically oriented representation, typically connecting content words and more suitable for relation extraction and shallow language understanding tasks.

In a similar vein and following closely the SD strategy, in Evalita 2014 different variants of the ISDT resource are exploited. The basic and *collapsed/propagated* representation variants are used in DPIE, whereas CLaP is based on the basic representation variant only. To obtain the *collapsed/propagated* version of ISDT, as well as the participants output, a CoNLL–to–RELS converter was implemented, whose result consists in a set of relations represented as triplets, i.e. name of the relation, governor and dependent. Note that following the SD approach, conjunct propagation is handled only partially by focusing on a limited and safe set of cases.

For CLaP, the Universal version of the basic ISDT variant (henceforth referred to as "uISDT") was used, annotated according to the Universal Stanford Dependencies scheme defined in the framework of *The Universal Dependency Treebank Project* [3]. uISDT was obtained through conversion from ISDT.

3 The Dependency Parsing for Information Extraction subtask

3.1 Task description

DPIE was organized as a classical dependency parsing task, where the performance of different parsers, possibly following different paradigms (statistical, rule-based, hybrid), can be compared on the basis of the same set of test data provided by the organizers.

In order to allow participants to develop and tune their systems, the ISDT resource was split into a training set (165,975 tokens) and a validation set (12,578 tokens). For the purposes of the final evaluation, we developed a new test data set, for a total of 9,442 tokens articulated into three subsets representative of different textual genres:

- a data set of 3,659 tokens extracted from newspaper texts and particularly rich in factual information, a feature making it suitable for evaluating Information Extraction capabilities (henceforth, IE–test)[4];

[2] See: http://medialab.di.unipi.it/wiki/ISDT

[3] https://code.google.com/p/uni-dep-tb/

[4] These texts are part of a benchmark used by Synthema

- a data set of 3,727 tokens from newspaper articles (henceforth, News–test);
- a data set of 2,056 tokens from European directives, annotated as part of the 2012 Shared Task on Dependency Parsing of Legal Texts (Dell'Orletta et al., 2012) (henceforth, SPLeT–test).

The main novelty of this task consists in the methodology adopted for evaluating the output of the participant systems. In addition to the Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS), which represent standard metrics in dependency parsing, we wanted to provide an alternative and semantically-oriented metric to assess the ability of the parsers to produce suitable and accurate output for information extraction applications. Whereas LAS and UAS were computed against the basic SD variant, represented in the CoNLL format, the semantically-oriented evaluation was computed against the *collapsed and propagated* version of the parsers output and was based on a subset of the relation types selected as more relevant, i.e. semantically-loaded.

The dependency relations that were selected for the semantically-oriented evaluation are 18 out of the 41 dependency types, namely: `acomp`, `advcl`, `advmod`, `amod`, `ccomp`, `dobj`, `iobj`, `mark`, `nn`, `nnp`, `npadvmod`, `nsubj`, `nsubjpass`, `prep`, `rcmod`, `tmod`, `vmod`, `xcomp`. Most of them link content words. In this case, used evaluation metrics are: *Precision*, the fraction of correct relations extracted over the total of extracted relations; *Recall*, the fraction of correct relations extracted over the relations to be found (according to the gold standard); and F1, the harmonic mean of the two.

Participants were allowed to use external resources, whenever they deemed it necessary, and to submit multiple runs. In the following section, we describe the main features of the participants' systems, together with achieved results.

3.2 Systems description and results

For DPIE, four participants submitted their results. Here follows an overview of the main features of their parsing systems[5], in order to provide a key to interpret the results achieved.

(`http://www.synthema.it/`) on a common project and kindly offered for the task.

[5]For a detailed description of each participant's system, please refer to the corresponding technical report.

Table 1 summarizes the main features of participants systems, based on three main parameters: 1) whether a single parser or a parser combination has been used; 2) the approach adopted by the parser (statistical, rule-based or hybrid), and 3) whether only the training and development sets provided by the organizers (DPIE only) or rather external resources (Other) have been used.

Participants mostly used publicly available state-of-the-art parsers and used them in different combinations for the task. The parsers that have been used are:

- MALT parser (Nivre et al., 2006): a transition–based dependency parser written in Java, which uses a SVM classifier;
- DeSR parser (Attardi et al., 2009): a transition–based dependency parser written in C++, which can be used with several classifiers including a Multi–Layer Perceptron;
- MATE parser (Bohnet, 2010): the MATE tools, written in Java, include both a graph-based parser and a transition-based parser. The transition-based MATE takes into account complete structures as they become available to re-score the elements of a beam, combining the advantages of transition–based and graph–based approaches. Efficiency is gained through Hash Kernels and exploiting parallelism.
- TurboParser (Martins et al., 2013): a C++ package that implements graph-based dependency parsing exploiting third-order features.
- ZPar (Zang and Nivre, 2011): a transition-based parser that leverages its performance by using considerably richer feature representations with respect to other transition-based parsers. It supports multiple languages and multiple grammar formalisms, but it was especially tuned for Chinese and English.

We provide below a short description of the parsing solutions adopted by each participant.

Attardi et al. (University of Pisa) The final runs submitted by this team used a combination of four parsers: MATE in the standard graph-based configuration; DeSR, with the Multilayer Perceptron algorithm; a new version of the DeSR parser, introducing graph completion; TurboParser.

Participant	#Parser/s used	Approach	Development
Attardi et al.	Combination	Statistical	DPIE only
Lavelli	Combination	Statistical	DPIE only
Mazzei	Combination	Statistical	DPIE only
Grella	Single	Hybrid	Other

Table 1: Systems overview based on number of parsers, approach and resources used.

Parser combination was based on the technique described in Attardi, Dell'Orletta (2009). Submitted runs differ at the level of the conversion applied to the corpus, performed in pre- and a post-processing steps, consisting in local restructuring of the parse-trees.

Lavelli (FBK-irst) This participant used the following parsers: ZPar; the graph-based MATE parser combined with the output of TurboParser (full model) using stacking; Ensemble (Surdenau and Manning, 2010), a parser that implements a linear interpolation of several linear-time parsing models. For the submission, the output of the following 5 parsers have been combined: graph-based MATE parser, transition-based MATE parser, TurboParser (full model), MaltParser (Nivre's arc-eager, PP-head, left-to-right), and MaltParser (Nivre's arc-eager, PP-head, right-to-left).

Mazzei (University of Torino) The final runs submitted by this participant resulted from the combination of the following parsers: MATE; DeSR parser with the Multi-Layer Perceptron algorithm; MALT parser.
Parser combination was based on the technique described in (Mazzei and Bosco, 2012), which applies a majority vote algorithm.

Grella (Parsit, Torino) This participant used a proprietary transition-based parser (ParsIt) based on a Multi-Layer Perceptron algorithm. The parser includes PoS tagging and lemmatization, using a dictionary of word forms with associated PoS, lemmas and morphology, and a subcategorization lexicon for verbs, nouns, adjectives and adverbs. In addition, the parser exploits a vectorial semantic space obtained by parsing large quantities of text with a basic parser. The parser was trained on a set of around 7,000 manually-annotated sentences, different from the ones provided for the task, and the output was converted into the ISDT scheme with a rule-based converter. The development resources were used in order to develop and test the converter from the output parser format into the ISDT representation format.

Tables 2 and 3 report the results for each run submitted by each participant system for the first evaluation track. In Table 2, the overall performance of parsers is reported in terms of achieved LAS/UAS scores, without considering punctuation. Since achieved results were very close for most of the runs, we checked whether the difference in performance was statistically significant by using the test proposed by Dan Bikel[6]. We considered that two runs differ significantly in performance when the computed p value is below 0.05. This was done by taking the highest LAS score and assessing whether the difference with subsequent values was significant or not; the highest score among the remaining ones whose difference was significant was taken as the top of the second cluster. This was repeated until the end of the list of runs. In Table 2, we thus clustered together the LAS of the runs whose difference was not significant according to the Bikel's test: the top results include all runs submitted by Attardi et al. and one of the runs by Lavelli.

Table 3 reports the performance results for each subset of the test corpus, covering different textual genres. It can be noticed that the best results are achieved with newspaper texts, corresponding to the IE and News test sets: in all runs submitted by participants higher results are obtained with the IE-test, whereas with the News-test LAS/UAS scores are slighly lower. As expected, for all participants the worse results refer to the test set represented by legal texts (SPLeT).

The results of the alternative and semantically-oriented evaluation, computed against the *collapsed* and *propagated* version of the systems out-

[6]The Randomized Parsing Comparator, whose script is now available at: `http://pauillac.inria.fr/~seddah/compare.pl`

Participant	LAS	UAS
Attardi run1	87.89	90.16
Attardi run3	87.84	90.15
Attardi run2	87.83	90.06
Lavelli run3	87.53	89.90
Lavelli run2	87.37	89.94
Mazzei run1	87.21	89.29
Mazzei run2	87.05	89.48
Lavelli run1	86.79	89.14
Grella	84.72	90.03

Table 2: DPIE subtask: participants' results, according to LAS and UAS scores. Results are clustered on the basis of the statistical significance test.

	IE	News	SPLeT
Attardi run1	88.64	87.77	86.77
Attardi run3	88.29	88.25	86.33
Attardi run2	88.55	88.09	86.01
Lavelli run3	88.71	87.68	85.21
Lavelli run2	88,8	87,29	84,99
Mazzei run1	88,2	87,64	84,71
Mazzei run2	88,2	86,94	85,21
Lavelli run1	87,72	87,39	84,1
Grella	86,96	84,54	81,08

Table 3: Systems results in terms of LAS on different textual genres.

put, are reported in Table 4, where Precision, Recall and F1 score for the set of selected relations are reported for each participant's run. In this case we did not perform any test of statistical significance. By comparing the results reported in tables 2 and 4, it is interesting to note differences at the level of the ranking of achieved results: besides the 3 runs by Attardi et al. which are top-ranked in both cases although with a different internal ordering, two runs by Mazzei (run2) and Lavelli (run1) respectively from the second cluster in table 2 show higher precision and recall than e.g. run3 by Lavelli which was among the top-ranked ones. The reasons underlying this state of affairs should be further investigated. It is however interesting to report that traditional parser evaluation with attachment scores (LAS/UAS) may not be always helpful for researchers who want to find the most suitable parser for their IE application, as suggested among others by Volokh and Neumann (2012).

We also performed a dependency–based evaluation, in order to identify low scored relations shared by all parsers. It turned out that `iobj` (indirect object), `nn` (noun compound modifier), `npadvmod` (noun phrase as adverbial modifier), `tmod` (temporal modifier) are hard to parse relations for all parsers, although at a different extent: their average F1 score computed on the best run of each participant ranges between 46,70 (`npadvmod`) and 56,25 (`tmod`). This suggests that either we do not have enough information for dealing with semantically–oriented distinctions (as in the case of `iobj`, `npadvmod` and `tmod`), or more simply the dimension of the training corpus is not sufficient to reliably deal with them (see the `nn` relation whose frequency of occurrence in Italian is much lower than in English).

Participant	Precision	Recall	F1
Attardi run1	81.89	90.45	85.95
Attardi run3	81.54	90.37	85.73
Attardi run2	81.57	89.51	85.36
Mazzei run2	80.47	89.98	84.96
Lavelli run1	80.30	88.93	84.39
Mazzei run1	80.88	87.97	84.28
Lavelli run2	79.13	87.97	83.31
Grella	80.15	85.89	82.92
Lavelli run3	78.28	88.09	82.90

Table 4: DPIE subtask: participants' results, according to Precision, Recall and F1 score of selected relations, computed against the *collapsed* and *propagated* variant of the output.

4 The Cross-Language dependency Parsing subtask

CLaP is a cross-lingual transfer parsing task, organized along the lines of the experiments described in McDonald et al. (2013). In this task, participants were asked to use their parsers trained on the Universal variant of ISDT (uISDT) on test sets of other languages, annotated according to the Universal Dependency Treebank Project guidelines. The languages involved in the task are all the

languages distributed from the Universal Dependency Treebank Project with the exclusion of Italian, i.e.: Brazilian-Portuguese, English, Finnish, French, German, Indonesian, Japanese, Korean, Spanish and Swedish.

Participant systems were provided with:

- a development set consisting of uISDT, the universal version of ISDT used for training in DPIE and obtained through automatic conversion, and validation sets of about 7,500 tokens for each of the eleven languages of the Universal Dependency Treebank;
- a number of test sets (one for each language to be dealt with) for evaluation, with gold PoS and morphology and without dependency information; these data sets consist of about 7,500 tokens for each of the eleven languages of the Universal Dependency Treebank. Test sets were built by randomly extracting sentences from SD treebanks available at `https://code.google.com/p/uni-dep-tb/`. For languages which opted for the Content Head (CH) option in the treatment of copulas, sentences with copular constructions were discarded.

The use of external resources (e.g. dictionaries, lexicons, machine translation outputs, etc.) in addition to the corpus provided for training was allowed. Participants in this task were also allowed to focus on a subset of languages only.

4.1 System description and results

Just one participant, Mazzei, submitted the system results for this task. He focused on four languages only: Brazilian-Portuguese, French, German and Spanish.

Differently from the approach previously adopted, for CLaP Mazzei used a single parser, the MALT parser. The adopted strategy is articulated in three steps as follows: 1) each analyzed test set was word-for-word translated into Italian using Google Translate; 2) the best feature configuration was selected for each language using MaltOptimizer (Ballesteros, 2012) on the translated development sets; 3) for each language the parsing models were obtained by combining the Italian training set with the translated development set.

Table 5 reports the results in terms of LAS, UAS and also LA (Label Accuracy Score). Unlike DPIE, the punctuation is included in the evaluation metrics.

	LAS	**UAS**	**LA**
Brazilian-Portuguese	71.70	76.48	84.50
French	71.53	77.30	84.41
German	66.51	73.86	79.14
Spanish	72.39	77.83	83.30

Table 5: CLaP results in terms of LAS, UAS, LA on the test sets.

The reported results confirm that using training data from different languages can improve accuracy of a parsing system on a given language: this can be particularly useful for improving the accuracy of parsing less–resourced languages. As expected, the accuracy achieved on the German test set is the lowest: typologically speaking, within the set of languages taken into account German is the most distant language from Italian. These results can be considered in the framework of the work proposed by Zhao et al. (2009), in which the authors translated word-for-word the training set in the target language: interestingly, Mazzei followed the opposite approach and achieved promising results.

5 Acknowledgements

Roberta Montefusco implemented the scripts for producing the collapsed and propagated version of ISDT and for the evaluation of systems in this variant. Google and Synthema contributed part of the resources that were distributed to participants.

References

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi and Joseph Turian. 2009. Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In *Proceedings of the 2nd Workshop of Evalita 2009*, Springer–Verlag, Berlin Heidelberg.

Giuseppe Attardi and Felice Dell'Orletta. 2009. Reverse Revision and Linear Tree Combination for Dependency Parsing. In *Proceedings of Human Language Technologie* (NAACL 2009), ACL, pp. 261–264.

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: An Optimization Tool for MaltParser. In *Proceedings of the System Demonstration Session of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics* (EACL 2012). ACL, pp. 58–62.

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics* (COLING 2010). ACL, pp. 89–97.

Cristina Bosco and Alessandro Mazzei. 2013. The EVALITA Dependency Parsing Task: from 2007 to 2011. In Bernardo Magnini, Francesco Cutugno, Mauro Falcone and Emanuele Pianta (eds.) *Evaluation of Natural Language and Speech Tools for Italian*, Springer–Verlag, Berlin Heidelberg, pp. 1–12.

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2012. Harmonization and merging of two Italian dependency treebanks. In *Proceedings of the LREC Workshop on Language Resource Merging*, ELRA, pp. 23–30.

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2013. Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In *Proceedings of the 7th ACL Linguistic Annotation Workshop and Interoperability with Discourse*, ACL, pp. 61–69

Felice DellOrletta, Simone Marchi, Simonetta Montemagni, Barbara Plank, Giulia Venturi. 2012. The SPLeT-2012 Shared Task on Dependency Parsing of Legal Texts. In *Proceedings of the 4th Workshop on Semantic Processing of Legal Texts (SPLeT 2012)*, held in conjunction with LREC 2012, Istanbul, Turkey, 27th May, pp. 42-51.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning 2006. Generating typed dependency parses from phrase structure parses. *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC 2006), ELRA, pp. 449–454.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford Typed Dependencies representation. In *Coling2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, CrossParser 08*, ACL, pp. 1–8 .

Marie-Catherine de Marneffe and Christopher D. Manning 2008. *Stanford Typed Dependencies manual* (Revised for the Stanford Parser v. 3.3 in December 2013). `http://nlp.stanford.edu/software/dependencies_manual.pdf`.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non–projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), ACL, pp. 617–622.

Alessandro Mazzei, and Cristina Bosco. 2012. Simple Parser Combination. In *Proceedings of Semantic Processing of Legal Texts (SPLeT-2012)*, ELRA, pp. 57–61.

Ryan T. McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B. Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of 51st annual meeting of the Association for Computational Linguistics* (ACL'13), ACL, pp. 92-97.

Simonetta Montemagni, Francesco Barsotti, Marco Battista, Nicoletta Calzolari, Alessandro Lenci, Ornella Corazzari, Antonio Zampolli, Francesca Fanciulli, Maria Massetani, Roberto Basili, Remo Raffaelli, Maria Teresa Pazienza, Dario Saracino, Fabio Zanzotto, Fabio Pianesi, Nadia Mana and Rodolfo Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillè (ed.) *Building and Using syntactically annotated corpora*, Kluwer, Dordrecht, pp. 189–210.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC '06), ELRA, pp. 2216–2219.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, Yi Zhang. 2014. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Dublin, Republic of Ireland, pp. 63–72.

Maria Simi, Cristina Bosco and Simonetta Montemagni. 2014. Less is More? Towards a Reduced Inventory of Categories for Training a Parser for the Italian Stanford Dependencies. In *Proceedings of the 9th International Conference on Language Resources and Evaluation* (LREC'14), ELRA, pp. 83–90.

Mihai Surdeanu and Christopher D. Manning. 2010. Ensemble Models for Dependency Parsing: Cheap and Good. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ACL, pp. 649–652.

Alexander Volokh and Günter Neumann. 2012. Task–oriented dependency parsing evaluation methodology. In Proceedings of the *IEEE 13th International Conference on Information Reuse & Integration, IRI*, Las Vegas, NV, USA, August 8-10, 2012, pp. 132–137.

Yue Zhang and Joakim Nivre. 2011. Transition-based Dependency Parsing with Rich Non-local Features. In *Proceedings of ACL*, ACL, pp. 188-193.

Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In *Proceedings of ACL/IJCNLP*, ACL, pp. 55-63.

10.12871/clicit201422

Dependency Parsing Techniques for Information Extraction

Giuseppe Attardi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
I-56127 Pisa, Italy
attardi@di.unipi.it

Maria Simi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
I-56127 Pisa, Italy
simi@di.unipi.it

Abstract

English. Dependency parsing is an important component in information extraction, in particular when using suitable formalisms and accurate and efficient parsing techniques. We review recent advances in dependency parsing and describe our own contribution in the context of the Evalita 2014 DPIE task.

Italiano. *Il parsing a dipendenze è un componente importante nell'estrazione di informazione da testi, in particolare quando usato con una rappresentazione adeguata e tecniche di parsing accurate ed efficienti. Accenniamo agli sviluppi recenti nel parsing a dipendenze e descriviamo il nostro contributo nel contesto del task DPIE di Evalita 2014.*

1 Introduction

Information extraction is one of the primary goals of text analytics. Text analytics is often performed by means of advanced statistical tools, relying on patterns or matching with gazetteers for identifying relevant elements from texts. Dependency parsing in an attractive technique for use in information extraction because it can be performed efficiently, parsers can be trained on treebanks in different languages, without having to produce grammars for each of them and they provide a representation that is convenient to use in any further layers of analysis.

The effectiveness of the dependency representation was shown for example in the CoNLL 2008 Shared task on Joint Dependency Parsing and Role Labelling (Surdeanu et al. 2008): over 80% of the roles did indeed correspond to either direct or double indirect dependency links. Stanford Dependencies (SD) introduce a notation for dependencies that is closer to the representation of the roles so that they are easier to extract. Universal Dependencies in particular, generalized from SD, are helpful for dealing uniformly with multiple languages (De Marneffe et al., 2014).

Deep parsing (Ballesteros et al., 2014) can extract "deep-syntactic" dependency structures from dependency trees that capture the argumentative, attributive and coordinative relations between full words of a sentence.

Practical uses of text analysis based on dependency structure are reported in many applications and domains, including medical, financial or intelligence. Google for example applies dependency parsing to most texts it processes (Goldberg, 2013): parse trees are used in extracting relations to build the Knowledge Vault (Dong et al., 2014) and to guide translation (Katz-Brown et al., 2011).

There is still potential for improving dependency parsers in several directions:

- Integration with other layers of analysis, e.g. POS tagging and role labelling.
- Improving the accuracy.
- Exploiting distributed word representations (word embeddings).

Recent work on improving accuracy has explored two issues: the strategy adopted in the analysis and the use of features in the parsing decision process.

Transitions parsers are affected by the problem of having to decide sometimes too early which attachment to make, before having seen the remaining part of the sentence.

Goldberg and Elhadad (2010) proposed a so-called "easy first" approach, directing the parser to complete the simplest structures first and dealing with their combination later when more information from the constituents is available.

Sartorio, Satta and Nivre (2013) propose new parsing rules that allow delaying attachments: e.g. given the two top stack words *w* and *z*, *RA-k* allows adding a dependency link from the *k*-th rightmost descendant of *w* to *z*. These parsing rules only handle cases of non-projectivity.

A similar effect can be obtained by using in a creative way the rules for handling non-projectivity introduced by Attardi (2006). The effect of *RA-k* can be obtained by delaying attachments performing *Shift*'s and recovering later using a *Left-k* rule, in cases where the delay turns out to have been unnecessary. This approach allows retaining the parser ability to handle non-projectivity.

During training, a parser is typically shown only one sequence of decoding actions computed by a training oracle guide that knows the correct parse tree. However there can be more than one sequence for building the same parse tree. Hence during training, the oracle could present all of them to the parser. This would teach the parser actions that may be useful in situations where it must recover from earlier errors.

These experimental solutions have still to find their way into a production dependency parser.

Besides the mentioned approach by Attardi for handling non-projectivity, another approach has been proposed later, which consists in introducing a single *Swap* action to exchange the two top elements of the stack. Often though the action though must be applied multiple times during parsing to move a whole constituent, one word at a time, to a new place where it can be eventually reduced. For example, the sentence:

```
Martin Marietta Corp. said it
won a $ 38.2 million contract
from the U.S. Postal Service to
manufacture and install auto-
mated mail - sorting machines .
```

requires the following sequence of actions[1]:

```
S R S L S R S S R S S S L S L S
R R S S S S S R R R L S swap S
S swap S S swap S S swap L L S
S swap S S swap S S swap L S S
swap R S S swap R R L L L L S L
L S L L
```

Basically, after the parser has reduced the phrases "`a $ 38.2 million contract`" and "`from the U.S. Postal Service`", it has to move the prepositional phrase "`to manufacture and install automated mail - sorting machines`" in front of the latter, by means of a sequence of alternating *Shift*/*Swap*, before it can be attached to the noun "`contract`". Nivre, Kuhlmann and Hall (2009) propose to handle this problem with an oracle that delays swaps as long as possible.

With the rules by Attardi (2006) instead, a single non-projective action (*Left*-2) is required to parse the above sentence:

```
S R S L S R S S R S S S L S L S
R R S S S S S R R R L L-2 S S S
S S L L S S S L S R S R R L L L
L L S L L
```

Notice that action *Left*-2 is equivalent to the pair *Swap RA*.

Non-projectivity has been considered a rare phenomenon, occurring in at most 7% of words in free order languages like Czech: however, counting the number of sentences, it occurs e.g. in over 60% of sentences in German.

Other approaches to deal with wrong too early parsing decision are to use a stacking combination of a left-to-right and right-to-left parser or to use a larger size beam. In the latter approach many alternative parsing are carried along and only later the wrong ones are pruned. Bohnet and Kuhn (2012) propose this approach in combination with a way to score the partial parse trees exploiting graph-based features.

Among the approaches to provide semantic word knowledge to improve parsing accuracy we mention the use of word clusters by Koo, Carreras and (2008) and leveraging information from the Knowledge Graph (Gesmundo and Hall, 2014). Word embeddings are used in the parser by Chen and Manning (2014).

2 Tools

Our experiments were based on DeSR, the first transition based parser capable of dealing directly with non-projective parsing, by means of specific non-projective transition rules (Attardi, 2006).

The DeSR parser is highly configurable: one can choose which classifier (e.g. SVM or Multi-Layer Perceptron) and which feature templates to use, and the format of the input, just by editing a configuration file. For example, to implement stacking, one needs to specify that the format of the input used by the second parser contains additional columns with the hints from the first par-

[1] We use a shorthand notation where `R` is a right parse action (aka *LA*), `L` is a left parse action (aka *RA*) and `S` is a *Shift*.

ser and how to extract features from them with a suitable feature model.

Rich features of the type proposed by Zhang and Nivre (2011) can be specified with the following notation, where 0 identifies the next token and -1 the last token, expressions indicate a path on the tree and eventually which token attribute to extract as a feature:

```
POSTAG(0)LEMMA(leftChild(-1))
```

It is also possible to represent conditional features, which depend on the presence of other words. For example, the following rule creates a pair consisting of the lemma of the next token and the lemma of the last token which was a verb, but only if the current token is a preposition:

```
if(POSTAG(0) = "E", LEMMA(0))
LEMMA(last(POSTAG, "V"))
```

Features may consist of portions of attributes that are selected by matching a regular expression. For example, a feature can be extracted from the morphology of a word:

```
match(FEATS(-1), "gen=.")
```

Binned distance features can be expressed as follows:

```
dist(leftChild(-1), 0))
```

Data Set

The EVALITA 2014 evaluation campaign on Dependency Parsing for Information Extraction is based on version 2.0 of the Italian Stanford Dependency Treebank (ISDT) (Bosco et al., 2013). It was provided to the participants split into a training set consisting of 7,398 sentences (158,447 tokens) and a development set of 580 sentences (12,123 tokens).

ISDT adopts an Italian variant of the Stanford Dependencies annotation scheme.

Experiments

The flexibility of DeSR allowed us to perform a number of experiments.

As a baseline we used DeSR MLP, which obtained scores of 87.36 % LAS and 89.64 % UAS on the development set. We explored using a larger number of features. However, adding for example 16 word-pair features and 23 triple-word features, the score dropped to 85.46 % LAS and 87.99 % UAS.

An explanation of why rich features are not effective with the DeSR parser is that it employs a Multi-Layer Perceptron that already incorporates non linearity in the second layer by means of a *softsign* activation function. Other parsers instead, which use linear classifier like perceptron or MIRA, benefit from the use of features from pairs or triples of words, since this provides a form of non-linearity.

To confirm this hypothesis, we built a version of DeSR that uses a passive aggressive perceptron and exploits graph completion, i.e. it also computes a graph score that is added to the cumulative transition score, and training uses an objective function on the whole sentence, as described in (Bohnet and Kuhn, 2012). This version of DeSR, called DeSR GCP, can still be configured providing suitable feature templates and benefits from reach features. In our experiments on the development set, it reached a LAS of 89.35%, compared to 86.48% of DeSR MLP.

2.1 Word Embeddings and Word Clusters

We explored adding some kind of semantic knowledge to the parser in a few ways: exploiting word embeddings or providing extra dictionary knowledge.

Word embeddings are potential conveyors of semantic knowledge about words. We produced word embeddings for Italian (IWE, 2014) by training a deep learning architecture (NLPNE, 2014) on the text of the Italian Wikipedia.

We developed a version of DeSR MLP using embeddings: a dense feature representation is obtained by concatenating the embedding for words and other features like POS, lemma and deprel, also mapped to a vector space. However, experiments on the development set did not show improvements over the baseline.

Alternatively to the direct use of embeddings, we used clusters of terms calculated using either the DBSCAN algorithm (Ester et al., 1996) applied to the word embeddings or directly through the word2vec library (WORD2VEC. 2014).

We added cluster features to our feature model, extracted from various tokens, but in no configuration we obtained an improvement over our baseline.

2.2 Adding transitivity feature to verbs

Sometimes the parser makes mistakes by exchanging subjects and passive subjects. This might have been due to its lack of knowledge about transitive verbs. We run an experiment by adding an extra attribute TRANS to verb tokens, denoting whether the verb is transitive, intransi-

tive or both. We added to the feature model the following rules:

```
if(POSTAG(0) = "V", TRANS(0))
LEMMA(-1)
if(POSTAG(-1) = "V", TRANS(-1))
LEMMA(0)
```

but the LAS on the development set dropped from 87.36 to 85.54.

2.3 Restructuring Parse Trees

Simi, Bosco and Montemagni (2014) argued for using a simpler annotation scheme than the ISDT schema. The proposed schema, called MIDT++, is attractive not just because of a smaller number of dependency types but also because it provides "easier to learn" dependency structures, which can be readily converted to ISDT.

The results from that paper suggested the idea of a transformational approach for the present DPIE task. We experimented performing several reversible transformations on the corpus, before training and after parsing.

The transformation process consists of the following steps:

1. apply conversion rules to transform the training corpus;
2. train a parser on the transformed training set;
3. parse the test sentences with the parser;
4. transform back the result.

Each conversion rule *Conv* must be paired with a $Conv^{-1}$ rule, for use in step 4, such that:

$$Conv^{-1}\ (Conv\ \mathrm{T}) = \mathrm{T}$$

for any dependency tree T. We tested the following transformations:

- *Conv-conj*: transform conjunctions from grouped (all conjuncts connected to the first one) to a chain of conjuncts (each conjunct connected to the previous one);
- *Conv-iobj*: for indirect objects, make the preposition the head, as it is the case for other prepositional complements;
- *Conv-prep-clauses*: for prepositional clauses, labeled either *vmod* or *xcomp*, make the preposition the head;
- *Conv-dep-clauses*: for subordinate clauses, *advcl* and *ccomp*, make the complementizer the head;
- *Conv-NNP*: turn proper nouns into a chain with the first token as head.

Arranging conjunctions in a chain is possibly helpful, since it reduces long-distance dependencies. The *Conv-conj* conversion however may entail a loss of information when a conjunct is in turn a conjunction, as for instance in the sentence:

Children applaud, women watch and smile …

In order to preserve the separation between the conjuncts, this transformation, and other similarly, introduce extra tags that allow converting back to the original form after parsing.

The transformations were quite effective on the development set, improving the LAS from 89.56% to 90.37%, but not as much on the official test set.

2.4 Parser configurations

In our final experiments we used the following parsers: transition-based DeSR MLP parser (Attardi et al., 2009), transition-based with graph completion DeSR GCP, graph-based Mate parser (Bohnet, 2010), graph-based TurboParser (Martin et al., 2012).

DESR MLP is a transition-based parser that uses a Multi-Layer Perceptron. We trained it on 320 hidden variables, with 40 iterations and a learning rate of 0.001, employing the following feature model:

Single word features
$s_2.l\ s_1.l\ b_0.l\ b_1.l\ b_2.l\ b_3.l\ b_0^{-1}.l\ lc(s_1).l\ lc(b_0).l\ rc(s_1).l\ rc(b_0).l$
$s_2.p\ s_1.p\ b_0.p\ b_1.p\ b_2.p\ b_3.p\ s_1^{+1}.p\ lc(s_1).p\ lc(b_0)$.p
$rc(s_1).p\ rc(b_0).p$
$s_1.c\ b_0.c\ b_1.c$
$s_1.m\ b_0.m\ b_1.m$
$lc(s_1).d\ lc(b_0).d\ rc(s_1).d$
$match(s_1.m$, "gen=.")
$match(b_0.m$, "gen=.")
Word pair features
$s_1.c\ b_0.c$
$b_0.c\ b_1.c$
$s_1.c\ b_1.c$
$s_1.c\ _2.c$
$s_1.c\ _3.c$
$rc(s_1).c\ b_0.c$
Conditional features
$if(b_0.p$ = "E", $b_0.l)\ last$(POSTAG, "V")).l

Table 1. Feature templates: s_i represents tokens on the stack, b_i tokens on the input buffer. $lc(s_i)$ and $rc(s_i)$ denote the leftmost and rightmost child of s_i, *l* denotes the lemma, *p* and *c* the POS and coarse POS tag, *m* the morphology, *d* the dependency label. An exponent indicates a relative position in the input sentence.

For the DeSR GCP parser we used the features described in (Bohnet and Nivre, 2012).

The Mate parser is a graph-based parser that uses passive aggressive perceptron and exploits reach features. The only configurable parameter is the number of iterations (set to 25).

TurboParser is a graph-based parser that uses third-order feature models and a specialized accelerated dual decomposition algorithm for making non-projective parsing computationally feasible. TurboParser was used in configuration "full", enabling all third-order features.

2.5 Parser combination

Further accuracy improvements are often achieved by ensemble combination of multiple parsers. We used the parser combination algorithm by Attardi and Dell'Orletta (2009), which is a fast linear algorithm and preserves a consistent tree structure in the resulting tree. This is relevant for the present task, since the evaluation is based on relations extracted from the tree. An algorithm that only chooses each link independently, based on independent voting, risks of destroying the overall tree structure.

3 Results

We submitted three runs, all with the same combination of the four parsers above. They differ only in the type of conversion applied to the corpus:

1. Run1: *Conv-iobj*, *Conv-prep-clauses*
2. Run2: no conversion
3. Run3: *Conv-iobj*, *Conv-prep-clauses, Conv-dep-clauses*

The first run achieved the best accuracy scores among all submissions, according to the LAS (Labeled Accuracy Score) and UAS (Unlabeled Accuracy Scores), as reported in Table 2. Punctuations are excluded from the evaluation metrics.

Run	LAS	UAS
Unipi_Run1	**87.89**	**90.16**
Unipi_Run2	87.83	90.06
Unipi_Run3	87.84	90.15

Table 2. Evaluation of accuracy on dependencies.

Unipi_Run1 also obtained the best scores in the evaluation of accuracy on extracted relations, as reported in Table 3.

The results show an apparent correlation between the two types of evaluations, which we observed consistently also during our experiments on the development set. Our tree-based combination algorithm preserves this property also on the combined output.

Run	Precision	Recall	F1
Unipi_Run1	**81.89**	**90.45**	**85.95**
Unipi_Run2	81.57	89.51	85.36
Unipi_Run3	81.54	90.37	85.73

Table 3. Evaluation on accuracy of relations.

The scores obtained on the test set are significantly lower than those we had obtained on the development set, where the same parser combination achieved 90.37% LAS and 92.54% UAS. Further analysis is required to explain such difference.

4 Conclusions

The Evalita 2014 task on Dependency Parsing for Information Extraction provided an opportunity to exploit a larger training resource for Italian, annotated according to an international standard, and to test the accuracy of systems in identifying core relations, relevant from the perspective of information extraction.

There have been significant advances recently in dependency parsing techniques, but we believe there are still margins for advances in the core techniques along two directions: new transition rules and strategies for applying them, and exploiting semantic information acquired from distributed word representations.

We have started exploring these ideas but for the moment, we achieved top accuracy in this task using just consolidated techniques.

These remain nevertheless promising research directions that are worth pursuing in order to achieve the performance and accuracy needed for large-scale information extraction applications.

Acknowledgments

Luca Atzori and Daniele Sartiano helped performing the experiments using embeddings and clusters.

References

Giuseppe Attardi. 2006. Experiments with a Multilanguage non-projective dependency parser. In: *Proc. of the Tenth Conference on Computational Natural Language Learning* (CoNLL-X '06), 166-170. ACL, Stroudsburg, PA, USA.

Giuseppe Attardi, Felice Dell'Orletta. 2009. Reverse Revision and Linear Tree Combination for Dependency Parsing. In: *Proc. of Human Language*

Technologies: The 2009 Annual Conference of the NAACL, Companion Volume: Short Papers, 261–264. ACL, Stroudsburg, PA, USA.

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi, Joseph Turian. 2009. Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In: *Proc. of Workshop Evalita 2009*, ISBN 978-88-903581-1-1.

Miguel Ballesteros, Bernd Bohnet, Simon Mille and Leo Wanner. 2014. Deep-Syntactic Parsing. In: *ProceedingsProc. of COLING 2014*.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proc. of Coling 2010*, pp. 89–97, Beijing, China. Coling 2010 Organizing Committee.

Bernd Bohnet and Jonas Kuhn. 2012. The Best of Both Worlds - A Graph-based Completion Model for Transition-based Parsers. In: *Proc. of EACL*. 2012, 77-87.

Bernd Bohnet and Joakim Nivre. 2012. Feature Description for the Transition-Based Parser for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing. Retrieved from http://stp.lingfil.uu.se/~nivre/exp/features_emnlp12.pdf

Cristina Bosco, Simonetta Montemagni, Maria Simi. 2013. Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In: *ACL Linguistic Annotation Workshop & Interoperability with Discourse*, Sofia, Bulgaria.

Danqi Chen and Christopher D. Manning. 2014. Fast and Accurate Dependency Parser using Neural Networks. In: *Proc. of EMNLP 2014*.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, Christopher D. Manning. 2014. Universal Stanford Dependencies: a Cross-Linguistic Typology. In: *Proc. LREC 2014*, Reykjavik, Iceland, ELRA.

Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion.

Martin Ester et al 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Int. Conference on Knowledge Discovery and Data Mining*. AAAI Press. pp. 226–231.

Andrea Gesmundo, Keith B. Hall. 2014. Projecting the Knowledge Graph to Syntactic Parsing. *Proc. of the 15th Conference of the EACL*.

Yoav Goldberg and Michael Elhadad. 2010. An Efficient Algorithm for Easy-First Non-Directional Dependency Parsing. *Proc. of NAACL-2010*.

Yoav Goldberg. 2013. Personal communication, http://googleresearch.blogspot.it/2013/05/syntactic-ngrams-over-time.html

IWE. 2014. Italian Word Embeddings. Retrieved from http://tanl.di.unipi.it/embeddings/.

Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, Masakazu Seno. 2011. Training a Parser for Machine Translation Reordering. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL 2008*, Columbus, Ohio, USA.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order nonprojective turbo parsers. In: *Proc. of the 51st Annual Meeting of the ACL* (Volume 2: Short Papers), 617–622, Sofia, Bulgaria. ACL.

Joakim Nivre, Marco Kuhlmann and Johan Hall. 2009. An Improved Oracle for Dependency Parsing with Online Reordering. *Proc. of the 11th International Conference on Parsing Technologies (IWPT)*, 73–76, Paris, October.

Ryan McDonald et al. 2013. Universal dependency annotation for multilingual parsing. In: *Proceedings of ACL 2013*.

NLPNET. 2014. Retrieved from https://github.com/attardi /nlpnet/

Maria Simi, Cristina Bosco, Simonetta Montemagni. 2008. Less is More? Towards a Reduced Inventory of Categories for Training a Parser for the Italian Stanford Dependencies. In: *Proc. LREC 2014*, 26–31, May, Reykjavik, Iceland, ELRA.

Mihai Surdeanu, Richard Johansson, Adam Meyers. Lluís Màrquez and Joakim Nivre, 2008. The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies, *Proc. of the 12th Conference on Computational Natural Language Learning*, 159–177, Manchester, August 20.

Francesco Sartorio, Giorgio Satta and Joakim Nivre. 2013. A Transition-Based Dependency Parser Using a Dynamic Parsing Strategy. In: *Proc. of ACL 2013.*

WORD2VEC. 2014. Retrieved from http://code.google.com/p/word2vec/

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In: *Proc. of the 49th ACL: Human Language Technologies*: Short papers, Volume 2, 188-193. ACL.

10.12871/clicit201423

Comparing State-of-the-art Dependency Parsers for the EVALITA 2014 Dependency Parsing Task

Alberto Lavelli
FBK-irst
via Sommarive, 18 - Povo
I-38123 Trento (TN) - ITALY
lavelli@fbk.eu

Abstract

English. This paper describes our participation in the EVALITA 2014 Dependency Parsing Task. In the 2011 edition we compared the performance of MaltParser with the one of an ensemble model, participating with the latter. This year, we have compared the results obtained by a wide range of state-of-the-art parsing algorithms (MaltParser, the ensemble model made available by Mihai Surdeanu, MATE parsers, TurboParser, ZPar). When evaluated on the development set according to the standard measure (i.e., Labeled Accuracy Score, LAS), three systems have obtained results whose difference is not statistically significant. So we have decided to submit the results of the three systems at the official competition. In the final evaluation, our best system, when evaluated according to LAS, ranked fourth (with a score very close to the best systems), and, when evaluated on the Stanford Dependencies, ranked fifth. The efforts reported in this paper are part of an investigation on how simple it is to apply freely available state-of-the-art dependency parsers to a new language/treebank.

Italiano. *Questo articolo descrive la partecipazione al Dependency Parsing Task a EVALITA 2014. Nell'edizione 2011 avevamo confrontato le prestazioni di MaltParser con un ensemble model, partecipando con quest'ultimo. Quest'anno abbiamo confrontato i risultati ottenuti da un insieme di algoritmi di parsing allo stato dell'arte (MaltParser, l'ensemble model di Mihai Surdeanu, i MATE parser, TurboParser, ZPar). Valutati sul development set in base alla misura standard (Labeled Accuracy Score, LAS), tre sistemi hanno ottenuto risultati le cui differenze non sono statisticamente significativi. Così abbiamo deciso di sottomettere i risultati dei tre sistemi alla competizione. Nella valutazione ufficiale, il nostro miglior sistema è risultato quarto, valutato in base a LAS (con un valore molto vicino a quello dei migliori sistemi) ed è risultato quinto, valutato in base alle Stanford Dependency. Gli sforzi riportati in questo articolo sono parte di un'indagine su quanto è facile applicare analizzatori sintattici a dipendenza liberamente disponibili a una nuova lingua / treebank.*

1 Introduction

Recently, there has been an increasing interest in dependency parsing, witnessed by the organisation of a number of shared tasks, e.g. Buchholz and Marsi (2006), Nivre et al. (2007). Concerning Italian, there have been tasks on dependency parsing in all the editions of the EVALITA evaluation campaign (Bosco et al., 2008; Bosco et al., 2009; Bosco and Mazzei, 2011). In the 2014 edition, the task on dependency parsing exploits the Italian Stanford Dependency Treebank (ISDT), a new treebank featuring an annotation based on Stanford Dependencies (de Marneffe and Manning, 2008).

This paper reports the efforts involved in applying several state-of-the-art dependency parsers for comparing their performance and participating in the EVALITA 2014 task on dependency parsing. Apart from participating in the EVALITA 2014 task, a second motivation was to investigate how simple is to apply freely available state-of-the-art dependency parsers to a new language/treebank following the instructions available together with the code and possibly having a few interactions

with the developers (Lavelli, 2014).

As in many other NLP fields, there are very few comparative articles when the performance of different parsers is compared. Most of the papers simply present the results of the newly proposed approach and compare them with the results reported in previous articles. In other cases, the papers are devoted to the application of the same tool to different languages/treebanks.

It is important to stress that the comparison concerns tools used more or less out of the box and that the results cannot be used to compare specific characteristics like: parsing algorithms, learning systems, ...

2 Description of the Systems

The choice of the parsers used in this study started from the two we already applied at EVALITA 2011, i.e. MaltParser and the ensemble method described by Surdeanu and Manning (2010). We then identified a number of other dependency parsers that in the last years have shown state-of-the-art performance, that are freely available and with the possibility of training on new treebanks. The ones included in the preliminary comparison reported in this paper are the MATE dependency parsers, TurboParser, and ZPar. In the near future, we plan to include other dependency parsers in our comparison. We have not been able to exploit some of the dependency parsers because of lack of time and some others because of different reasons: they are not yet available online, they lack documentation on how to train the parser on new treebanks (the ClearNLP dependency parser), they have limitations in the encoding of texts (input texts only in ASCII and not in UTF-8; the Redshift dependency parser), ...

MaltParser (Nivre et al., 2006) (version 1.8) implements the transition-based approach to dependency parsing, which has two essential components:

- A nondeterministic transition system for mapping sentences to dependency trees
- A classifier that predicts the next transition for every possible system configuration

Given these two components, dependency parsing can be performed as greedy deterministic search through the transition system, guided by the classifier. With this technique, it is possible to perform parsing in linear time for projective dependency trees and quadratic time for arbitrary (non-projective) trees (Nivre, 2008). MaltParser includes different built-in transition systems, different classifiers and techniques for recovering non-projective dependencies with strictly projective parsers.

The ensemble model made available by Mihai Surdeanu (Surdeanu and Manning, 2010)[1] implements a linear interpolation of several linear-time parsing models (all based on MaltParser). In particular, it combines five different variants of MaltParser (Nivre's arc-standard left-to-right, Nivre's arc-eager left-to-right, Covington's non projective left-to-right, Nivre's arc-standard right-to-left, Covington's non projective right-to-left) as base parsers. Each individual parser runs in its own thread, which means that, if a sufficient number of cores are available, the overall runtime is essentially similar to a single MaltParser. The resulting parser has state-of-the-art performance yet it remains very fast.

The MATE tools[2] include both a graph-based parser (Bohnet, 2010) and a transition-based parser (Bohnet and Nivre, 2012; Bohnet and Kuhn, 2012). For the languages of the 2009 CoNLL Shared Task, the graph-based MATE parser reached accuracy scores similar or above the top performing systems with fast processing. The speed improvement is obtained with the use of Hash Kernels and parallel algorithms. The transition-based MATE parser is a model that takes into account complete structures as they become available to rescore the elements of a beam, combining the advantages of transition-based and graph-based approaches.

TurboParser (Martins et al., 2013)[3] (version 2.1) is a C++ package that implements graph-based dependency parsing exploiting third-order features.

ZPar (Zhang and Nivre, 2011) is a transition-based parser implemented in C++. ZPar supports multiple languages and multiple grammar formalisms. ZPar has been most heavily developed for Chinese and English, while it provides generic support for other languages. It leverages a global discriminative training and beam-search

[1] http://www.surdeanu.info/mihai/ensemble/

[2] https://code.google.com/p/mate-tools/

[3] http://www.ark.cs.cmu.edu/TurboParser/

			collapsed and propagated		
	LAS		P	R	F_1
MATE stacking (TurboParser)	89.72		82.90	90.58	86.57
Ensemble (5 parsers)	89.72		82.64	90.34	86.32
ZPar	**89.53**		84.65	92.11	88.22
MATE stacking (transition-based)	89.02		82.09	89.77	85.76
TurboParser (model_type=full)	88.76		83.32	90.71	86.86
TurboParser (model_type=standard)	88.68		83.07	90.55	86.65
MATE graph-based	88.51		81.72	89.42	85.39
MATE transition-based	88.32		80.70	89.40	84.82
Ensemble (MaltParser v.1.8)	88.15		80.69	88.34	84.34
MaltParser (Covington non proj)	87.79		81.50	87.39	84.34
MaltParser (Nivre eager -PP head)	**87.53**		81.30	88.78	84.88
MaltParser (Nivre standard - MaltOptimizer)	86.35		81.17	89.04	84.92
Ensemble (MaltParser v.1.3)	86.27		78.57	86.28	82.24

Table 1: Results on the EVALITA 2014 development set without considering punctuation. The second column reports the results in term of Labeled Attachment Score (LAS). The score is in bold if the difference with the following line is statistically significant. The three columns on the right show the results in terms of Precision, Recall and F_1 for the collapsed and propagated relations.

			collapsed and propagated		
	LAS		P	R	F_1
MATE stacking (transition-based)	87.67		79.14	88.14	83.40
Ensemble (5 parsers)	87.53		78.28	88.09	82.90
MATE stacking (TurboParser)	87.37		79.13	87.97	83.31
MATE transition-based	87.07		78.72	87.16	82.73
MATE graph-based	86.91		78.74	87.97	83.10
ZPar	86.79		80.30	88.93	84.39
TurboParser (model_type=full)	86.53		79.43	89.42	84.13
TurboParser (model_type=standard)	86.45		79.65	89.32	84.21
Ensemble (MaltParser v.1.8)	85.94		76.30	86.38	81.03
MaltParser (Nivre eager -PP head)	**85.82**		78.47	86.06	82.09
Ensemble (MaltParser v.1.3)	85.06		76.36	84.74	80.33
MaltParser (Covington non proj)	84.94		77.24	82.97	80.00
MaltParser (Nivre standard - MaltOptimizer)	84.44		76.53	86.99	81.43

Table 2: Results on the EVALITA 2014 test set without considering punctuation. The second column reports the results in term of Labeled Attachment Score (LAS). The score is in bold if the difference with the following line is statistically significant. The three columns on the right show the results in terms of Precision, Recall and F_1 for the collapsed and propagated relations.

framework.

2.1 Experimental Settings

The level of interaction with the authors of the parsers varied. In two cases (ensemble, MaltParser), we have mainly exploited the experience gained in previous editions of EVALITA. In the case of the MATE parsers, we have had a few interactions with the author who suggested the use of some undocumented options. In the case of TurboParser, we have simply used the parser as it is after reading the available documentation. Concerning ZPar, we have had a few interactions with the authors who helped solving some issues.

As for the ensemble, at the beginning we repeated what we had already done at EVALITA 2011 (Lavelli, 2011), i.e. using the ensemble as it is, simply exploiting the more accurate extended models for the base parsers. The results were unsatisfactory, because the ensemble is based on an old version of MaltParser (v.1.3) that performs worse than the current version (v.1.8). So we decided to apply the ensemble model both to the output produced by the current version of MaltParser and to the output produced by some of the parsers used in this study. In the latter case, we have used the output of the following 5 parsers: graph-based MATE parser, transition-based MATE parser, TurboParser (full model),

			collapsed and propagated		
	LAS		P	R	F_1
Ensemble (5 parsers)	87.22		78.21	87.92	82.78
MATE stacking (transition-based)	**86.99**		78.42	87.70	82.80
MATE transition-based	86.47		78.08	87.11	82.35
ZPar	86.40		79.84	88.27	83.84
TurboParser (model_type=full)	86.35		79.77	89.12	84.19
MATE graph-based	86.34		77.94	87.02	82.23
TurboParser (model_type=standard)	86.32		79.50	89.39	84.16
MATE stacking (TurboParser)	85.87		76.79	86.43	81.32
Ensemble (MaltParser v.1.8)	85.87		76.59	86.58	81.28
MaltParser (Nivre eager -PP head)	**85.66**		78.28	86.89	82.36
MaltParser (Covington non proj)	84.98		77.24	83.24	80.13
Ensemble (MaltParser v.1.3)	84.75		75.52	83.98	79.52
MaltParser (Nivre standard - MaltOptimizer)	84.25		76.29	86.77	81.19

Table 3: Results on the EVALITA 2014 test set after training on the training set only (NO development set) without considering punctuation. The second column reports the results in term of Labeled Attachment Score (LAS). The score is in bold if the difference with the following line is statistically significant. The three columns on the right show the results in terms of Precision, Recall and F_1 for the collapsed and propagated relations.

MaltParser (Nivre's arc-eager, PP-head, left-to-right), and MaltParser (Nivre's arc-eager, PP-head, right-to-left).

Concerning MaltParser, in addition to using the best performing configurations at EVALITA 2011[4], we have used MaltOptimizer[5] (Ballesteros and Nivre, 2014) to identify the best configuration. According to MaltOptimizer, the best configuration is Nivre's arc-standard. However, we have obtained better results using the configurations used in EVALITA 2011. We are currently investigating this issue.

As for the MATE parsers, we have applied both the graph-based parser and the transition-based parser. Moreover, we have combined the graph-based parser with the output of another parser (both the transition-based parser and TurboParser) using stacking. Stacking is a technique of integrating two parsers at learning time[6], where one of the parser generates features for the other.

Concerning ZPar, the main difficulty was the fact that a lot of RAM is needed for processing long sentences (i.e., sentences with more than 100 tokens need 70 GB of RAM). After some interactions with the authors, we were able to understand and fix this issue.

During the preparation of the participation in the task, the experiments were performed using the split provided by the organisers, i.e. training on the training set and testing using the development set.

When applying stacking, we have performed 10-fold cross validation of the first parser on the training set, using the resulting output to provide to the second parser the predictions used during learning. During parsing the output of the first parser (trained on the whole training set and applied to the development set) has been provided to the second parser.

3 Results

In Table 1 we report the parser results on the development set ranked according to decreasing Labeled Accuracy Score (LAS), considering punctuation. The score is in bold if the difference with the following line is statistically significant[7] (the difference is significant only if p-value is less than 0.05). In the three columns on the right of the table the results for the collapsed and propagated relations are shown (both the conversion and the evaluation are performed using scripts provided by the organisers).

In Table 1 we have grouped together the parsers if the differences between their results (in terms of

[4]Nivre's arc-eager, PP-head, and Covington non projective.

[5]http://nil.fdi.ucm.es/maltoptimizer/

[6]Differently from what is done by the ensemble method described above where the combination takes place only at parsing time.

[7]To compute the statistical significance of the differences between results, we have used MaltEval (Nilsson and Nivre, 2008).

LAS) are not statistically significant. As it can be seen, five clusters can be identified.

Note that the computation of the statistical significance of the results was possible only for the standard evaluation (LAS) but not for the evaluation of the recognition of Stanford Dependencies. This is obviously a strong limitation in the possibility of analysing the results. We plan to investigate if it is possible to perform such computation.

An obvious remark is that the ranking of the results according to LAS and according to the recognition of Stanford Dependencies is different. This made the choice of the parsers for the participation difficult, given that the participants would have been ranked based on both measures.

According to the results on the development set, we decided to submit for the official evaluation three models: ZPar, MATE stacking (TurboParser), and the ensemble combining 5 of the best parsers. For the official evaluation, the training was performed using both the training and the development set. In Table 2. you may find the results of all the parsers used in this study (in italics those submitted to the official evaluation). Comparing Table 1 and Table 2, it emerges that some of the parsers show different behaviours between the development and the test set. This calls for an analysis to understand the reasons of such difference. The results of a preliminary analysis are reported in Section 4.

The results obtained by the best system submitted to the official evaluation are: 87.89 (LAS), 81.89/90.45/85.95 (P/R/F_1). According to LAS, our systems were ranked fourth (the ensemble combining 5 of the best parsers), fifth (MATE parser stacking based on TurboParser) and eighth (ZPar). Evaluating using Stanford Dependencies was different. The same systems were ranked ninth, seventh, and fifth respectively. More details about the task and the results obtained by the participants are available in Bosco et al. (2014).

4 Discussion

We are currently analysing the results shown above to understand how to further proceed in our investigation. A general preliminary consideration is that, as expected, approaches that combine the results of different parsers perform better than those based on a single parser model, usually with the drawback of a bigger complexity.

The results shown in Tables 1 and 2 raise a few questions.

The first question concern the fact that some of the parsers (e.g., ZPar) show different behaviours between the development and the test set. This is still true even if we consider the clusters of where the results are not statistically different. To investigate this issue we performed some experiments training on the training set only (not using the development set) and analysing the test set. These results are reported in Table 3. The results show that some parsers have different behaviours on the development set and on the test set, even when considering only the clustering performed taking into account the statistical significance of the difference between different parsers' performance. This issue needs to be further investigated.

The second question concern the discrepancy between the standard evaluation in terms of LAS and the recognition of the Stanford dependencies in terms of Precision, Recall and F_1. For example, the ensemble is our best scoring system according to the standard evaluation, while is our worst system when evaluated on the Stanford dependencies. A crucial element to investigate this issue is the possibility of computing the statistical significance of the difference between the results of the recognition of Stanford Dependencies.

5 Conclusions and Future Work

In the paper we have reported on work in progress on the comparison between several state-of-the-art dependency parsers on the Italian Stanford Dependency Treebank (ISDT) in the context of the EVALITA 2014 dependency parsing task.

In the near future, we plan to widen the scope of the comparison including more parsers and analysing some unexpected behaviours emerged from our experiments.

Finally, we will perform an analysis of the results obtained by the different parsers considering not only their performance but also their behaviour in terms of speed, CPU load at training and parsing time, ease of use, licence agreement, ...

Acknowledgments

This work was partially supported by the EC-funded project EXCITEMENT (FP7ICT-287923). We wish to thank the authors of the parsers for making them freely available. In particular, we would like to thank Bernd Bohnet, Joakim Nivre, Mihai Surdeanu, Yue Zhang, and Yijia Liu for

kindly answering our questions on the practical application of their parsers and for providing useful suggestions.

References

Miguel Ballesteros and Joakim Nivre. 2014. MaltOptimizer: Fast and effective parser optimization. *Natural Language Engineering*, FirstView:1–27, 10.

Bernd Bohnet and Jonas Kuhn. 2012. The best of both worlds – a graph-based completion model for transition-based parsers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 77–87, Avignon, France, April. Association for Computational Linguistics.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island, Korea, July. Association for Computational Linguistics.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China, August. Coling 2010 Organizing Committee.

Cristina Bosco and Alessandro Mazzei. 2011. The EVALITA 2011 parsing task: the dependency track. In *Working Notes of EVALITA 2011*, pages 24–25.

Cristina Bosco, Alessandro Mazzei, Vincenzo Lombardo, Giuseppe Attardi, Anna Corazza, Alberto Lavelli, Leonardo Lesmo, Giorgio Satta, and Maria Simi. 2008. Comparing Italian parsers on a common treebank: the EVALITA experience. In *Proceedings of LREC 2008*.

Cristina Bosco, Simonetta Montemagni, Alessandro Mazzei, Vincenzo Lombardo, Felice DellOrletta, and Alessandro Lenci. 2009. Evalita09 parsing task: comparing dependency parsers and treebanks. In *Proceedings of EVALITA 2009*.

Cristina Bosco, Felice Dell'Orletta, Simonetta Montemagni, Manuela Sanguinetti, and Maria Simi. 2014. The EVALITA 2014 dependency parsing task. In *Proceedings of EVALITA 2014*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City, June. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK, August. Coling 2008 Organizing Committee.

Alberto Lavelli. 2011. An ensemble model for the EVALITA 2011 dependency parsing task. In *Working Notes of EVALITA 2011*.

Alberto Lavelli. 2014. A preliminary comparison of state-of-the-art dependency parsers on the italian stanford dependency treebank. In *Proceedings of the first Italian Computational Linguistics Conference (CLiC-it 2014)*.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jens Nilsson and Joakim Nivre. 2008. MaltEval: an evaluation and visualization tool for dependency parsing. In *Proceedings of LREC 2008*.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 2216–2219.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic, June. Association for Computational Linguistics.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553.

Mihai Surdeanu and Christopher D. Manning. 2010. Ensemble models for dependency parsing: Cheap and good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 649–652, Los Angeles, California, June. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June. Association for Computational Linguistics.

10.12871/clicit201424

Testing parsing improvements with combination and translation in Evalita 2014

Alessandro Mazzei
Dipartimento di Informatica
Università degli Studi di Torino
Corso Svizzera 185, 10149 Torino
mazzei@di.unito.it

Abstract

English. We present the two systems used by the UniTo group to participate to the Evalita 2014 parsing tasks. In particular, we describe the ensemble parser system used for DPIE task and the parsing-by-translation system used for the CLaP task.

Italiano. *Presentiamo i due sistemi utilizzati dal gruppo UniTo per partecipare alla competizione sul parsing di Evalita 2014. Descriviamo il sistema di ensamble parsing usato nel DPIE task e il sistema basato su traduzione usato per partecipare al CLaP task.*

1 Introduction

In the last years a great attention has been devoted to the dependency formalisms and parsers (Kübler et al., 2009). As a consequence many research lines follow new techniques in order to improve the parsing performances, e.g. (Carreras, 2007; Surdeanu and Manning, 2010). However, the specific applicative scenario can draw a clear playground where improvements can be effectively measured. The Evalita 2014 competition on parsing set up two distinct parsing tasks: (1) the Dependency Parsing for Information Extraction (DPIE) task, and (2) the Cross-language Dependency Parsing (CLaP) task.

The DPIE task is the "classical" dependency parsing task for the evaluation of the parsing systems on the Italian language (Bosco and Mazzei, 2012). However, in contrast with the previous editions of the task, the DPIE task adopts the new ISDT treebank (Bosco et al., 2013), which is based on the stanford dependency annotation (de Marneffe and Manning, 2008b), and uses two distinct evaluation measures: the first is the traditional LAS (Labeled Attachment Score), the second is related to the Information Extraction process and is based on a subset of the dependency relations inventory.

The CLaP task wants to test the utility of a standard cross-lingual annotation schema in order to parse foreign languages. By using an universal variant (McDonald et al., 2013) of the Italian ISDT treebank (U-ISDT) as learnin set, one has to parse sentences of several foreign languages.

In order to participate to both the tasks we devised two distinct parsing systems. We participate to the DPIE task by reusing a very simple ensamble parsing system (Mazzei and Bosco, 2012) (Section 2), and we participate to the CLaP task by designing a new cross-language parsing system that uses an on-line translator as external knowledge source (Section 3).

2 The DPIE task

The Dependency Parsing for Information Extraction (DPIE) is the main task of EVALITA 2014 competition on parsing. The focus is on standard dependency parsing of Italian texts. The evaluation is performed on two directions: the LAS (Labelled Attachment Score) as well as a measure on the *collapsed propagated dependencies*, i.e. on simple transformations of a subset of the whole dependency set, which usually are expressed in form of triples (de Marneffe and Manning, 2008a). In particular, the measure based on collapsed propagated dependencies is designed to test the utility of the dependency parsing with respect to the general process of Information Extraction.

In order to participate to this task we decided to reuse the system described in (Mazzei and Bosco, 2012), which follows two promising directions towards the improvement of the performance of the statistical dependency parsers. Indeed, some new promising parsing algorithms use larger sets of syntactic features, e.g. (McDonald and Pereira, 2006; Carreras, 2007), while others apply gen-

eral techniques *to combine* together the results of various parsers (Zeman and Žabokrtskỳ, 2005; Sagae and Lavie, 2006; Hall et al., 2007; Attardi and dell'Orletta, 2009; Surdeanu and Manning, 2010; Lavelli, 2012). We explored both these directions in our participation to the DPIE task by combining three state of the art statistical parsers. The three parsers are the MATE[1] parser (Bohnet, 2010) (version 3.61), the DeSR[2] parser (Attardi, 2006) (version 1.4.3), the MALT[3] parser (Nivre et al., 2006) (version 1.7.2). We combined these three parsers by using two very simple voting algorithms (Breiman, 1996; Zeman and Žabokrtskỳ, 2005), on the standard configurations for learning and classification.

The MATE parser (Bohnet, 2009; Bohnet, 2010) is a development of the algorithms described in (Carreras, 2007), and it basically adopts the second order maximum spanning tree dependency parsing algorithm. In particular, Bohnet exploits *hash kernel*, a new parallel parsing and feature extraction algorithm that improves the accuracy as well as the parsing speed (Bohnet, 2010).

The DeSR parser (Attardi, 2006) is a transition (shift-reduce) dependency parser similar to (Yamada and Matsumoto, 2003). It builds dependency structures by scanning input sentences in left-to-right and/or right-to-left direction. For each step, the parser learns from the annotated dependencies if to perform a shift or to create a dependency between two adjacent tokens. DeSR can use different set of rules and includes additional rules to handle non-projective dependencies. The parser can choose among several learning algorithms (e.g Multi Layer Perceptron, Simple Vector Machine), providing user-defined feature models.

The MALT parser (Nivre et al., 2006) implements the transition-based approach to dependency parsing too. In particular MALT has two components: (1) a (non-deterministic) transition system that maps sentences to dependency trees; (2) a classifier that predicts the next transition for every possible system configuration. MALT performs a greedy deterministic search into the transition system guided by the classifier. In this way, it is possible to perform parsing in linear time for projective dependency trees and quadratic time for arbitrary (non-projective) trees.

[1] `http://code.google.com/p/mate-tools/`
[2] `http://sites.google.com/site/desrparser/`
[3] `http://maltparser.org/`

2.1 The combination algorithms

We combine the three parsers by using two very simple algorithms: COM1 (Algorithm 1) and COM2 (Algorithm 2), both implemented in the PERL programming language. These algorithms have been previously experimented in (Zeman and Žabokrtskỳ, 2005) and in (Surdeanu and Manning, 2010). The main idea of the COM1 algorithm

```
foreach sentence do
    foreach word W in the sentence S do
        if DepP2(W) == DepP3(W) then
            Dep-COM1(W) := DepP2(W)
        else
            Dep-COM1(W) := DepP1(W)
        end
    end
end
```

Algorithm 1: The combination algorithm COM1, that corresponds to the *voting* algorithm reported in (Zeman and Žabokrtskỳ, 2005)

is to do a democratic voting among the parsers. For each word in the sentence, the dependency (the parent and the edge label) assigned to the word by each parser is compared: if at least two parsers assign the same dependency, the COM1 algorithm selects that dependency. In the case that each parser assigns a different dependency to the word, the algorithm selects the dependency assigned by the *best parser*. As noted by (Zeman and Žabokrtskỳ, 2005), who use the name *voting* for COM1, this is the most logical decision if it is possible to identify a priori the best parser, in contrast to the more democratic random choice.

```
foreach sentence do
    foreach word W in the sentence S do
        if DepP2(W) == DepP3( W) then
            Dep-COM2(W) := DepP2(W)
        else
            Dep-COM2(W) := DepP1(W)
        end
    end
    if TREE-COM2(S) is corrupted then
        TREE-COM2(S) := TREE-P1(S)
    end
end
```

Algorithm 2: The combination algorithm COM2, that corresponds to the *switching* algorithm reported in (Zeman and Žabokrtskỳ, 2005)

	MATE	DeSR	MALT	COM1	COM2
DevSet	89.65	86.19	86.26	89.60	89.65
TestSet	87.05	84.15	84.61	87.21	87.05

Table 1: The LAS score for the MATE, DeSR and MALT parsers, their simple combinations COM1 and COM2 on the development and test sets.

The COM2 algorithm is a simple variation of the COM1. COM1 is a single word combination algorithm that does not consider the whole dependency structure. This means that incorrect dependency trees can be produced by the COM1 algorithm: cycles and multiple roots can destroy the *treeness* of the structure. The solution that we adopt in the COM2 algorithm is quite naive: if the tree produced by the COM1 algorithm for a sentence is corrupted, then the COM2 returns the tree produced by the best parser. Again, similarly to (Zeman and Žabokrtskỳ, 2005), who use the name *switching* for COM2, this is the most logical decision when there is an emerging best parser from a development data set.

2.2 Experimental Results

We applied our approach for parsing combination in two stages. In the first stage we use the development set to evaluate the best parser and in the second stage we use the COM1 and COM2 algorithms to parse the test set. For all the experiments we used two machines. A powerful Linux workstation, equipped with 16 cores, processors 2GHz, and 128 GB ram has been used for the training of the MATE and Malt parsers. Morever, we have not been able to install DeSR on this machine, so we use a virtual Linux workstation equipped with a single processor 1GHz, and 2 GB ram has been used DeSR. The MALT and DeSR parsers accept as input the CONLL-07 format, that is the format provided by the task organizers. In contrast, MATE accepts the CONLL-09 format: simple conversions scripts have been implemented to manage this difference.

A first run was performed in order to evaluate the best parser in the COM1 and COM2 algorithms with respect to the LAS. We used the ISDT training ($file\ isdt\_train.conll$, $165,975$ words) as training set and the ISDT development ($file$: $isdt\_devel.conll$, $12,578$ words) as development set. The first row in Table 1 shows the results of the three parsers in this first experiment. MATE parser outperforms the DeSR and MALT parsers of $\sim 3\%$ better. On the basis of this result, we used MATE as our best parser in the combination algorithms (cf. Section 2.1).

COM1 and COM2 reach the score of 89.60% and 89.65% respectively. So, on the development set there is no improvement on the performance of the best parser. The reason of this is evident from table 2, that details the results of the three parsers on the development set on the basis of their agreements. The second row of this table show that when $DeSR == MALT! = MATE$, the combination algorithm gives the *wrong* selection preferring the majority.

In a second run, we used the union of the training and development set as a whole training set ($files$: $isdt\_train.conll, isdt\_devel.conll$) and we used the blind file provided by the organizers as test set ($file$: $DPIE\_Test\_DS\_blind.conll$, $9,442$ words). The second row in Table 1 shows the results of the three parsers in this second experiment: the LAS values 87.21% and 87.05%, produced by COM1 and COM2, are the official results for of our participation to the DPIE task.

There is a $\sim 0.15\%$ difference between the COM1 and COM2 results and in Table 3 we detailed the results of the three parsers on the test set. When the three parsers agree on the same dependency (Table 3, first row), this happens on $\sim 80.27\%$ of the words, they have a very high LAS score, i.e. 94.03%. In contrast to the development set, DeSR and MALT parsers do better than the MATE parser only when they agree on the same dependency (Table 3, second row). The inspection of the other rows in Table 3 shows that COM1 algorithms has the best possible performance w.r.t. the voting strategy. Finally, the fact that COM2 produces the same result of MATE shows that the LAS improvement produces always a non-correct tree in the final output.

In Table 4 we report the results of the system with respect to the measure defined on the propagated and collapsed dependencies. In contrast to the LAS measure, here COM1 produces a worse result than COM2. So, improvements in the LAS

	MATE	DeSR	MALT	COM1	COM2
DevSet	84.8/92.0/88.2	80.7/89.2/84.7	81.0/89/0/84.8	85.2/91.2/88.1	84.8/92.0/88.2
TestSet	80.5/90.0/85.0	76.9/86.7/81.5	76.8/86.6/81.4	80.9/88.0/**84.3**	80.5/90.0/**85.0**

Table 4: The collapsed and propagated dependency sore in terms of precison/recall/F-score for the collapsed dependencies for the three parsers, their simple combinations (COM1 and COM2) on the development and test sets.

					%
MATE 95.4	==	DeSR	==	MALT	**81.8**
MATE 43.5	!=	**DeSR 39.8**	==	**MALT**	**4.9**
MATE	==	**DeSR 70.9**	!=	MALT 13.1	**4.8**
MATE	==	**MALT 70.0**	!=	DeSR 15.6	**5.0**
MATE 46.6	!=	DeSR 10.9	!=	MALT 15.5	**3.6**

Table 2: The detailed performances on the LAS score of the three parsers and their simple combination on the ISDT development set. Note that we are computing the scores with punctuation.

					%
MATE 94.03	==	DeSR	==	MALT	**80.28**
MATE 40.7	!=	**DeSR 41.9**	==	**MALT**	**5.34**
MATE	==	**DeSR 62.2**	!=	MALT 19.4	**5.11**
MATE	==	**MALT 67.4**	!=	DeSR 17.6	**5.25**
MATE 35.9	!=	DeSR 15.9	!=	MALT 17.8	**4.03**

Table 3: The detailed performances on the LAS score of the three parsers and their simple combination on the ISDT test set. Note that we are computing the scores with punctuation.

produces as drawback a decline with respect to this measure.

3 The CLaP task

The Cross-language Dependency Parsing (CLaP) is a pilot task focusing on cross-lingual transfer parsing. In this subtask it is asked to learn from the Italian Stanford Dependency Treebank annotated in with the universal dependencies ($file$: $isdt\_udl.conll$), and to test on sentences of other languages (McDonald et al., 2013). In particular, we decided to participate to the task on four specific languages: German (DE), Espanol (ES), French (FR) and Brazilian Portuguese (PT-BR). For each language, the organizers provided a development file.

In CLaP task we used only one parser, i.e. the MALT parser. We decided to use this parser since there is a related system, called MaltOptimizer (Ballesteros and Nivre, 2012) (version 1.0.3), that allows for a straight optimization of the various parameters of the MALT parser. Indeed, our strategy was to train the MALT parser on the universal isdt by using the specific algorithm and features which optimize the learning on the development set of the target language. Moreover, in order to supply lexical information to the parsing algorithm, we used *Google_translate* (`https://translate.google.com`) to translate foreign words in Italian. In Figure 1 we reported the workflow adopted in this task for learning and parsing of the French language (it is analogous for the other languages). The learning stage is composed by five steps:

1. A script extracts the foreign words from the development set

2. Google_translate translates the foreign words, contained in one single file, into Italian.

3. A script recomposes the development set with Italian words

4. MaltOptimizer uses the recomposed development set in order to produce a configuration file (algorithm and features).

5. The MALT parser uses the configuration file to produce a parsing model file.

In a similar way, the parsing stage is composed by five steps:

1. A script extracts the foreign words from the test set.

2. Google_translate translates the foreign words,

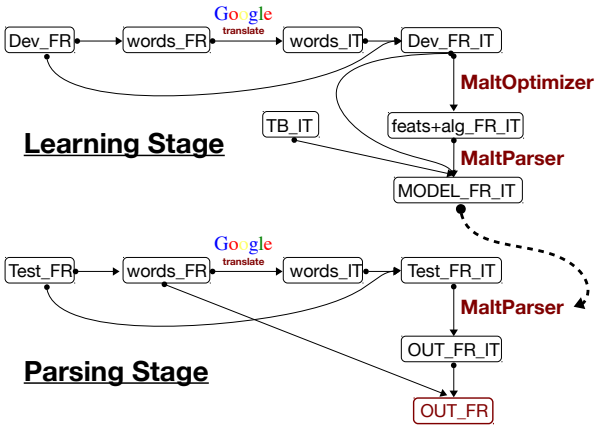


Figure 1: The workflow adopted fot the CLaP task for the French language: the schema is identical for the Spanish, German, Brazilian-Portoguese.

	DE	ES	FR	PT-BR
Baseline 1	60.23	67.72	66.74	66.12
Baseline 2	66.51	71.69	71.60	71.70
System	66.51	72.39	71.53	71.70

Table 5: The LAS score for CLaP task on the test sets for German (DE), Espanol (ES), French (FR), Brazilian-Portoguese (PT-BR) languages.

contained in one single file, into Italian.

3. A script recomposes the test set with Italian words.

4. The MALT parser uses the parsing model to parse the recomposed test set.

5. A script recomposes the parsing test set with the foreign words.

In Table 5 we reported the results in terms of LAS measure of the system together with two baselines. The baseline 1 it has been produced by training the MALT parser with the standard configuration on the learning set obtained by the union of the u-ISDT with the original development set of the foreign language. The baseline 2 it has been produced by training the MALT parser with the standard configuration on the learning set obtained by the union of the u-ISDT with the translated development set of the foreign language. The results proves that our workflow produces an improvement on the LAS measure of $5 - 6\%$ for each language. Comparing the baselines, we can say that the improvements are essentially by the translation process rather than the optimization process.

4 Conclusions

In this paper we described the two systems used by the UniTo group to participate to EVALITA 2014 parsing competition. The first, used in the DPIE task, is a very simple ensamble parsing algorithm; the second is a cross-language parsing algorithm that uses an on-line translator as external knowledge source.

In the DPIE task, we can see that the performance of the ensamble system with respect to the bast parser is quite neglectable, in contrast to the results obtained in other competition (Mazzei and Bosco, 2012). This result suggests that the performance of the simple ensamble algorithms adopted are highly sensitive from the leaning set adopted.

In the CLaP task, we can see that the performance of the developed system outperforms the baseline for all the four languages. This result confirms the possibility to improve parsing performances by using data developed for other languages.

References

Giuseppe Attardi and Felice dell'Orletta. 2009. Reverse revision and linear tree combination for dependency parsing. In *HLT-NAACL*, pages 261–264.

Giuseppe Attardi. 2006. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 166–170, New York City, June. Association for Computational Linguistics.

Miguel Ballesteros and Joakim Nivre. 2012. Maltoptimizer: A system for maltparser optimization. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Bernd Bohnet. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 67–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China, August. Coling 2010 Organizing Committee.

Cristina Bosco and Alessandro Mazzei. 2012. The evalita dependency parsing task: from 2007 to 2011. In *Evaluation of Natural Language and Speech Tools for Italian - Proceedings of Evalita 2011*, volume 7689, pages 1–12. Springer-Verlag, Heidelberg. ISBN: 978-3-642-35827-2.

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2013. Converting italian treebanks: Towards an italian stanford dependency treebank. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 61–69, Sofia, Bulgaria, August. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 957–961.

Marie-Catherine de Marneffe and Christopher D. Manning, 2008a. *Stanford typed dependencies manual*, September. http://nlp.stanford.edu/software/dependencies_manual.pdf.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008b. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryigit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? a study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939.

Sandra Kübler, Ryan T. McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Alberto Lavelli. 2012. An Ensemble Model for the EVALITA 2011 Dependency Parsing Task. In *Working Notes of EVALITA 2011*. CELCT a r.l. ISSN 2240-5186.

Alessandro Mazzei and Cristina Bosco. 2012. Simple Parser Combination. In *SPLeT 2012 – Fourth Workshop on Semantic Processing of Legal Texts (SPLeT 2012) – First Shared Task on Dependency Parsing of Legal Texts*, pages 57–61.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006))*, volume 6, pages 81–88.

Ryan T. McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B. Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *ACL (2)*, pages 92–97. The Association for Computer Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: a data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, volume 2216-2219.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In Robert C. Moore, Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, *HLT-NAACL*. The Association for Computational Linguistics.

Mihai Surdeanu and Christopher D. Manning. 2010. Ensemble models for dependency parsing: Cheap and good? In *NAACL*. The Association for Computational Linguistics.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, volume 3.

Daniel Zeman and Zdenek Žabokrtskỳ. 2005. Improving parsing accuracy by combining diverse dependency parsers. In *International Workshop on Parsing Technologies. Vancouver, Canada*, pages 171–178. Association for Computational Linguistics.

10.12871/clicit201425

EVENTI
EValuation of Events and Temporal INformation at Evalita 2014

Tommaso Caselli*
VU Amsterdam
De Boelelaan 1105, Amsterdam
t.caselli@gmail.com

Rachele Sprugnoli
FBK - University of Trento
Via Sommarive 18, Trento
sprugnoli@fbk.eu

Manuela Speranza
FBK
Via Sommarive 18, Trento
manspera@fbk.eu

Monica Monachini
ILC-CNR
Via G. Moruzzi 1, Pisa
monica.monachini@ilc.cnr.it

Abstract

English. This report describes the EVENTI (*EValuation of Events aNd Temporal Information*) task organized within the EVALITA 2014 evaluation campaign. The EVENTI task aims at evaluating the performance of Temporal Information Processing systems on a corpus of Italian news articles. Motivations for the task, datasets, evaluation metrics, and results obtained by participating systems are presented and discussed.

Italiano. *Questo report descrive il task EVENTI (EValuation of Events aNd Temporal Information) organizzato nell'ambito della campagna di valutazione EVALITA 2014. EVENTI mira a valutare le prestazioni dei sistemi di processamento automatico dell'informazione temporale su un corpus di articoli di giornale in lingua italiana. Le motivazioni alla base del task, i dataset, le metriche di valutazione ed i risultati ottenuti dai sistemi partecipanti sono presentati e discussi.*

1 Introduction

Temporal Processing has recently become an active area of research in the NLP community. Reference to time is a pervasive phenomenon of human communication, and it is reflected in natural language. Newspaper articles, narratives and other text documents focus on events, their location in time, and their order of occurrence. Text comprehension itself involves, in large part, the ability to identify the events described in a text, locate them in time (and space), and relate them according to their order of occurrence. The ultimate goal of a temporal processing system is to identify all temporal elements (events, temporal expressions and temporal relations) either in a single document or across documents and provide a chronologically ordered representation of this information. Most NLP applications, such as Summarization, Question Answering, and Machine Translation, will benefit from such a capability. The TimeML Annotation Scheme (Pustejovsky et al., 2003a) and the release of annotated data have facilitated the development of temporally aware NLP tools. Similarly to what has been done in other areas of NLP, five open evaluation challenges[1] have been organized in the area of Temporal Processing. TempEval-2 has also boosted multilingual research in Temporal Processing by making TimeML compliant data sets available in six languages, including Italian. Unfortunately, partly due to the limited size (less than 30,000 tokens), no system was developed for Italian. Before the EVENTI challenge, there was no complete system for Temporal Processing in Italian, but only independent modules for event (Robaldo et al., 2011; Caselli et al., 2011b) and temporal expressions processing (HeidelTime) (Strötgen et al., 2014).

The EVENTI evaluation exercise[2] builds upon

* Formerly at Trento RISE

[1] TempEval-1: http://www.timeml.org/tempeval/; TempEval-2 http://timeml.org/tempeval2/; TempEval-3 http://www.cs.york.ac.uk/semeval-2013/task1/; TimeLine http://alt.qcri.org/semeval2015/task4/, and QA TempEval http://alt.qcri.org/semeval2015/task5/

[2] https://sites.google.com/site/eventievalita2014/

previous evaluation campaigns to promote research in Temporal Processing for Italian by offering a complete set of tasks for comprehension of temporal information in written text. The exercise consists of a Main task on contemporary news and a Pilot task on historical texts and is based on the EVENTI corpus, which contains 3 datasets: the Main task training data, the Main task test data and the Pilot task test data.

2 EVENTI Annotation

The EVENTI exercise is based on the EVENTI annotation guidelines, a simplified version of the Italian TimeML Annotation Guidelines (henceforth, It-TimeML) (Caselli, 2010), using four It-TimeML tags: TIMEX3, EVENT, SIGNAL and TLINK. For clarity's sake, we report only the changes which have been applied to It-TimeML.

The TIMEX3 tag is used for the annotation of temporal expressions. No changes have been made with respect to It-TimeML.

The EVENT tag is used to annotate all mentions of events including verbs, nouns, prepositional phrases and adjectives. Changes concern the event extent. In particular, we have introduced exceptions to the minimal chunk rule for multi-token event expressions (the list of multi-token expressions created for this purpose is available online[3]). We have simplified the annotation of events realized by adjectives and prepositional phrases by restricting it to the cases in which they occur in predicate position with the explicit presence of a copula or a copular verb.

The SIGNAL tag identifies textual items which encode a relation either between EVENTs, or TIMEX3s or both. In EVENTI, we have annotated only SIGNALs indicating temporal relations.

The TLINK tag did not undergo any changes in terms of use and attribute values. Major changes concern the definition of the set of temporal elements that can be involved in a temporal relation. Details on this aspect are reported in the description of subtask C in Section 3.

3 EVENTI Subtasks

The EVENTI evaluation exercise is composed of a Main Task and a Pilot Task. Each task consists of a set of subtasks in line with previous TempEval campaigns and their annotation methodology.

The subtasks proposed are:

- Subtask A: determine the extent, the type and the value of temporal expressions (i.e. timex) in a text according to the It-TimeML TIMEX3 tag definition. For the first time, empty TIMEX3 tags were taken into account in the evaluation;

- Subtask B: determine the extent and the class of the events in a text according to the It-TimeML EVENT tag definition;

- Subtask C: identify temporal relations in raw text. This subtask involves performing subtasks A and B and subsequently identifying the pairs of elements (event - event and event - timex pairs) which stand in a temporal relation (TLINK) and classifying the temporal relation itself. Given that EVENTI is an initial evaluation exercise in Italian and to avoid the difficulties of full temporal processing, we have further restricted this subtask by limiting the set of candidate pairs to: i.) pairs of main events in the same sentence; ii.) pairs of main event and subordinate event in the same sentence; and iii.) event - timex pairs in the same sentence. All temporal relation values in It-TimeML are used; i.e. BEFORE, AFTER, IS_INCLUDED, INCLUDES, SIMULTANEOUS, I(MMEDIATELY)_AFTER, I(MMEDIATELY)_BEFORE, IDENTITY, MEASURE, BEGINS, ENDS, BEGUN_BY and ENDED_BY.

- Subtask D: determine the value of the temporal relation given two gold temporal elements (i.e. the source and the target of the relation) as defined in Task C (main event - main event; main event - subordinate event; event - timex).

4 Data Preparation and Distribution

The EVENTI evaluation exercise is based on the EVENTI corpus, which consists of 3 datasets: the Main task training data, the Main task test data and the Pilot task test data.

The news stories distributed for the Main task are taken from the Ita-TimeBank (Caselli et al., 2011a). Two expert annotators have conducted a manual revision of the annotations for the Main

[3] https://sites.google.com/site/eventievalita2014/data-tools/poliremEVENTI.txt

(a) *Event Class Values.* (b) *Temporal Relations Values.*

Figure 1: Distribution of event classes and temporal relations in the EVENTI corpus (in percent).

task to solve inconsistencies mainly focusing on harmonizing event class and temporal relation values. The annotation revision has been performed using CAT[4] (Bartalesi Lenzi et al., 2012), a general-purpose web-based text annotation tool that provides an XML-based stand-off format as output. The final size of the EVENTI corpus for the Main task is 130,279 tokens, divided in 103,593 tokens for training and 26,686 for test.

The Main task training data have been released to participants in two separate batches[5] through the Meta-Share platform[6]. Annotated data are available under the Creative Commons Licence Attribution-NonCommercial-ShareAlike 3.0 to facilitate re-use and distribution for research purposes.

The Pilot test data consist of about 5,000 tokens from newspaper articles published in "*Il Trentino*" by Alcide De Gasperi, one of the founders of the Italian Republic and one of the fathers of the European Union (De Gasperi, 2006). All the selected news stories date back to 1914, the year of the outbreak of World War 1, a topic particularly relevant in 2014, the 100th anniversary of the Great War. They have been manually annotated in CAT by an expert annotator who followed the EVENTI Annotation Guidelines. As the aim of the Pilot task was to analyze how well systems built for contemporary languages perform on historical texts, no training data have been provided and participants were asked to participate with the systems developed for the Main task.

[4] `http://dh.fbk.eu/resources/cat-content-annotation-tool`

[5] ILC Training Set: `http://goo.gl/3kPJkM`; FBK Training Set: `http://goo.gl/YnQWml`

[6] `http://www.meta-share.eu/`

	Main Training	Main Test	Pilot Test
EVENTs	17,835	3,798	1,195
TIMEX3s	2,735	624	97
SIGNALs	932	231	62
TLINKs	3,500	1,061	382

Table 1: Annotated events, temporal expressions, signals and temporal relations in the EVENTI corpus.

Table 1 reports the total number of each annotated element type in the Main task training set, in the Main task test set, and in the Pilot test set.

	Main Training	Main Test	Pilot Test
EVENTs	172.1	142.4	239
TIMEX3s	26.4	23.3	19.0
TLINKs	33.7	39.7	76.4

Table 2: Average number of annotated events, temporal expressions and temporal relations per 1,000 tokens in the EVENTI corpus.

Table 2 presents the comparison between the average number of EVENTs, TIMEX3s and TLINKs annotated in the three datasets. The Pilot corpus clearly shows a higher density of events (238 vs. 172.1 and 142.4 for training and test, respectively) and temporal relations (76.4 vs. 33.7 and 39.7 for training and test, respectively). On the other hand, the average number of temporal expressions in the two corpora is comparable.

We illustrate in Figure 1 the distribution of the class values of EVENTs and the distribution of the temporal values for TLINKs. We can observe an even distribution of all classes among the three datasets. The most frequent classes are OCCURRENCE and STATE, followed by I_STATE and I_ACTION. The high prevalence of occurrences

and states is not surprising as these classes encode the objects of a narrative (e.g. contemporary news or historical texts) or what people "speak about". On the other hand, more interesting results are provided by the relatively high presence of the I_STATE and I_ACTION classes. According to the TimeML definitions, these classes are used either to express intensional relations or speculations about "possible worlds" between events. They are markers of subjectivity along the axis of event factivity, pointing out that people do not limit themselves to "speak about" happenings but they also speculate on these happenings. The higher frequency of I_STATE in the Pilot corpus with respect to the Main datasets is due to the fact that the Pilot dataset is mainly composed of editorial comments which frequently contain perspectives on and speculations about the world by the writer. Additional evidence is also the lower frequency of the REPORTING class in the Pilot dataset than in the Main task. The high presence of personal opinions influences also the temporal structure of the texts whereby most events are not ordered chronologically but presented as belonging to the same time frame on top of which the author expresses his opinions and suggests future and alternative courses of events. As a matter of fact, the most frequent temporal relation in the Pilot task is SIMULTANEOUS. On the other hand, in the Main task there is an evident preference for IS_INCLUDED. The main task is composed of news articles where events tend to be more often linked to temporal containers (e.g. temporal expressions or other events) to facilitate understanding of stories by readers.

5 Evaluation

Given the strong connection of this task with the TempEval Evaluation Exercises, we adopted the evaluation metrics developed in TempEval-3 (UzZaman et al., 2013) with minor modifications[7]. In particular, the scorer was adapted in order to take CAT files as input and the evaluation of temporal expressions was extended to include empty TIMEX3 tags.

Concerning the temporal elements in subtask A and subtask B, we evaluated: i) the number of the elements correctly identified and if their extension is correct, and ii.) the attribute values correctly identified. For recognition, we used Precision, Recall and F1-score. Strict and relaxed match were both taken into account. As for attribute evaluation, we used F1-score to measure how well a system identifies an element and its attribute values. For subtask A, we computed Attribute F1-score on VALUE and Attribute F1-score on TYPE, and based the final ranking on the former. For subtask B, we computed attribute F1-score on CLASS, on which we based the final ranking.

For subtask C, we took into consideration three aspects : i) the number and the extent of the temporal elements identified in a raw text ii) the identification of the correct sources and targets applying both strict and relaxed match and iii) the identification of the correct temporal value. In subtask D, we evaluated only the identification of the correct temporal value. Similarly to subtasks A and B, we computed Precision, Recall and F1-score also for subtasks C and D and we set the final rankings on the basis of F-1 scores[8].

6 Participant Systems

Although eight teams registered for the task, only three actually submitted the output of their systems for a total of 17 unique runs: FBK (Fondazione Bruno Kessler), HT (University of Heidelberg), and UNIPI (Università di Pisa). We report below a short description of the systems the three teams developed. Detailed descriptions are reported in the system papers of the Evalita 2014 Proceedings (Bosco et al., 2014).

FBK is an end-to-end system based on a machine learning approach, namely supervised classification. It was developed for the EVENTI exercise by combining and adapting to Italian three subsystems first developed for English within the NewsReader project[9]: one for time expression recognition and normalization, one for event extraction, and one for temporal relation identification and classification. Temporal expression recognition and classification is conducted by means of an adaptation to Italian of TimeNorm (Bethard, 2013), a rule-based system based on synchronous context free grammars. The other subsystems are based on machine learning and use a Support Vector Machine approach.

HeidelTime is a rule-based, multilingual and

[7]The scorer of EVENTI is available online: `http://goo.gl/TbnE7D`

[8]TLINK directionality was not an issue as the scorer is able to deal with reciprocal temporal relations

[9]`http://www.newsreader-project.eu`

		RECOGNITION				NORMALIZATION	
		F1	P	R	Strict F1	TYPE F1	VALUE F1
MAIN TASK	HT 1.7	0.78	0.921	0.676	0.662	0.643	0.571
	HT 1.8	0.893	0.935	0.854	0.821	0.643	**0.709**
	HT 1.8 (no ET)	0.878	0.94	0.824	0.804	0.775	0.69
	FBK_A1	0.886	0.936	0.841	0.827	0.8	0.665
	UNIPI_1	0.768	0.929	0.654	0.662	0.643	0.566
	UNIPI_2	0.771	0.922	0.662	0.659	0.64	0.563
PILOT TASK	HT 1.7	0.653	0.96	0.495	0.585	0.571	0.408
	HT 1.8	0.788	0.918	0.691	0.671	0.624	0.459
	HT 1.8 (no ET)	0.781	0.917	0.68	0.663	0.615	0.45
	FBK_A1	0.87	0.963	0.794	0.746	0.678	**0.475**

Table 3: Results of Main and Pilot tasks for subtask A - TIMEX3s recognition and normalization.

		RECOGNITION				CLASS
		F1	P	R	Strict F1	F1
MAIN TASK	FBK_B1	0.884	0.902	0.868	0.867	**0.671**
	FBK_B2	0.749	0.917	0.632	0.732	0.632
	FBK_B3	0.875	0.915	0.838	0.858	0.67
PILOT TASK	FBK_B1	0.843	0.9	0.793	0.834	**0.604**
	FBK_B2	0.681	0.897	0.548	0.671	0.535
	FBK_B3	0.83	0.92	0.756	0.819	0.602

Table 4: Results of Main and Pilot tasks for subtask B - Events recognition and *class* assignment.

		F1	P	R	Strict F1
MAIN TASK	FBK_C1 (B1_D1)	**0.264**	0.296	0.238	0.341
	FBK_C2 (B1_D2)	0.253	0.265	0.241	0.325
	FBK_C3 (B2_D1)	0.209	0.282	0.167	0.267
	FBK_C4 (B2_D2)	0.168	0.203	0.255	0.258
	FBK_C5 (B3_D1)	0.247	0.297	0.211	0.327
	FBK_C6 (B3_D2)	0.247	0.297	0.211	0.327
PILOT TASK	FBK_C1 (B1_D1)	**0.185**	0.277	0.139	0.232
	FBK_C2 (B1_D2)	0.174	0.233	0.139	0.221
	FBK_C3 (B2_D1)	0.141	0.243	0.099	0.178
	FBK_C4 (B2_D2)	0.139	0.215	0.102	0.174
	FBK_C5 (B3_D1)	0.164	0.268	0.118	0.209
	FBK_C6 (B3_D2)	0.164	0.268	0.118	0.209

Table 5: Results of Main and Pilot tasks for subtask C - Temporal relations from raw texts.

cross-domain temporal tagger initially developed for English in the context of TempEval-2 (Strötgen and Gertz, 2010), which makes use of regular expressions. The distributed version of HeidelTime, which is freely available under a GNU General Public License, already supports Italian temporal tagging. For the EVENTI exercise, HT extended HeidelTime by tackling the recognition of TimeML's empty TIMEX3 tags and by tuning HeidelTime's Italian resources (e.g. by extending patterns, adding rules, and improving existing ones) on the basis of the more specific annotation guidelines and the training data released by the task organizers.

UNIPI used the available version of HeidelTime and adapted it by integrating into the pipeline the Tanl tools (Attardi et al., 2010), a suite of statistical machine learning tools for text analytics based on the software architecture paradigm of data pipelines.

7 System Results

For subtask A, temporal expression recognition and normalization, we had 3 participants and 6 unique runs. Table 3 shows the results for both the Main and the Pilot tasks. In the Main Task, only the best scoring run, i.e. HT 1.8, achieved results in terms of F1 above 0.70 in the normalization of the VALUE attribute. However, in the assignment of the TYPE attribute, FBK_A1 outperformed it (0.8 vs. 0.643). As for recognition, all the runs have a precision above 0.92, while recall ranges from 0.654 to 0.854. An analogous trend in the recognition of temporal expressions was registered in the Pilot task. The best run proved to be FBK_A1 with a VALUE F1 of 0.475.

Only one team participated in the remaining three subtasks. In subtask B, event detection and classification, 3 different runs were submitted. The evaluation results are reported in Table 4. FBK_B1 is the best run both in the Main task and in the Pilot task with an F1 on class assignment of 0.671 and 0.604 respectively. FBK_B1 has the best results also in terms of event recognition (0.884 in the Main task and 0.843 in the Pilot task). Precision in event recognition is high, above 0.89, in both tasks. Recall, on the other hand, ranges from 0.548, the lowest score obtained in the Pilot task, to 0.868, the highest score obtained in the Main task.

Results of Main and Pilot tasks for subtask C, i.e. temporal relations from raw texts, are reported in Table 5. For both Main task and Pilot task, the best performing run is FBK_C1, with 0.264 F-score and 0.185 F-score respectively.

In subtask D, i.e. TLINKs with temporal elements given, two runs were submitted. As shown in Table 6, FBK_D1 performed better than FBK_D2 with a difference of more than 0.3 points (0.736 vs. 0.419).

	F1	P	R	Strict F1
FBK_D1	**0.736**	0.74	0.731	0.731
FBK_D2	0.419	0.342	0.541	0.309

Table 6: Results of Main and Pilot tasks for subtask D - TLINKs with temporal elements given.

8 Discussion

EVENTI achieved a significant result in setting the state of the art on Temporal Processing for Italian although the reduced number of participants for three of the four subtasks limits observations on the participants' results.

Subtask A, temporal expression recognition and normalization, attracted the highest number of participants. Two participants, HT and UNIPI, developed rule-based systems both for recognition and normalization and submitted three and two runs respectively: HT 1.7 (the HT system publicly available), HT 1.8 (the system adapted to EVENTI) , HT 1.8 (the adapted system wothout the empty tag feature), UNIPI_1 (a baseline obtained by using the same publicly available system as HT 1.7), and UNIPI_2 (obtained substituting the TreeTagger with the Tanl Tokenizer in HeidelTime). FBK, on the other hand, developed a hybrid system: recognition is conducted by means of an SVM classifier while normalization is provided by a rule based system adapted to Italian (TimeNorm). Concerning recognition of temporal expressions, competition among the best performinig systems, HT 1.8 and FBK_A1, is high (the difference in performance is less than 1%). On the Main task data (contemporary news articles), the statistical system, FBK_A1, performs best at strict matching, and only one rule-based system, HT 1.8, performs best at relaxed matching. The difference in performance between the two rule based systems, HT and UNIPI_2, both for recognition and normalization clearly points to a problem in the integration of the Tanl POS tagset in the HT system, rather than signaling a limit of the approach for this task. Unfortunately, it is not possible to compare these results with those obtained by the systems participating in the EVALITA 2007 TERN (*Temporal Expression Recognition and Normalization*) Task (Bartalesi Lenzi and Sprugnoli, 2007) for two main reasons: firstly, the annotation of TIMEX3 tags substantially differs from that for TIMEX2, which was used for TERN, in terms of tag spans, normalization and presence of empty timex tags; and secondly, the evaluation methods in TERN, except for the recognition task, are not comparable with those used in EVENTI.

Subtask B, event detection and classification, had only one team with 3 different runs. The FBK system is based on an SVM classifier. The difference in performance between the three runs does not concern the features used for training but the classification method. The best result, FBK_B1's strict F1 0.867, was obtained by splitting the detection and classification task into two steps, first detection and then classification, and using a one-vs-one strategy. In the classification task, the predictions of the detection classifier were incorporated as a feature. FBK_B3, which obtained comparable results to FBK_B1, implements a single classifier with one-vs-rest multi-class classification. Difference in performance is less than 1% suggesting that both approaches are highly competitive but require different multi-class classification methods. Semantics is encoded by means of lexical knowledge through MultiWordNet (Pianta et al., 2002). Comparisons with (Caselli et al., 2011b) and (Robaldo et al., 2011) are not possible due to the different sizes of the training and

test sets and also because the original TempEval-2 test set for Italian has been incorporated in the EVENTI training set. Nevertheless, the results reported in (Caselli et al., 2011b) for event classes suggest that more fine grained and specialized lexical knowledge for event classification may provide better results.

Subtasks C and D are focused on temporal relations. The unique participant, i.e. FBK, submitted 6 runs for subtask C and 2 for subtask D. The system for subtask C tackles the task in a two step approach: first an SVM classifier identifies all eligible event-event and event-timex pairs for a temporal relation. Subsequently, a second SVM classifier, based on a previous framework for temporal relations between entities (Mirza and Tonelli, 2014), assigns the temporal relations values. This classifier mostly uses basic morphosyntactic features plus additional information based on the annotated SIGNAL. Different versions of the system (FBK_C2, FBK_C4, FBK_C6 and FBK_D2) incorporate TLINK rules for event-timex pairs which include signals as reported in the annotation guidelines. The system for subtask D corresponds to the second SVM classifier developed for subtask C. In both subtasks the presence of rules for event-timex temporal relations have a negative impact on system performance.

Concerning the Pilot task, no comparisons with previous evaluations can be drawn. To the best of our knowledge, EVENTI is the first evaluation exercise on Temporal Information Processing on historical texts. In general, a drop in the systems' performance was registered. In particular, the drop in the normalization of temporal expressions can probably be explained by the fact that 54% of the temporal expressions in the Pilot corpus is fuzzy (e.g. *i sacrifici dell'⟨ora presente⟩*) or non-specific (e.g. *nei ⟨giorni⟩ del dolore*), with respect to 24% in the Ita-TimeBank. A similar decrease in performance was registered in subtask D, submitted post evaluation by FBK, where both runs achieved an F1-score of 0.57.

8.1 Comparison with TempEval-3

Although no direct comparison can be made, it is still interesting to compare the performance among systems in different languages, developed and tested on annotation schemes which are compliant with a common standard (i.e. ISO-TimeML). We report in Table 7 the results of the best systems from TempEval-3 (UzZaman et al., 2013) for English (EN) and Spanish (ES) with respect to the identification of temporal relation from raw text.

		Strict F1	F1 attribute
TASK A	HT 1.8	0.893	0.709
	HeidelTime_EN	0.813	0.776
	HeidelTime_ES	0.853	0.875
TASK B	FBK_B1	0.867	0.671
	ATT-1_EN	0.810	0.718
	TIPSemB-F_ES	0.888	0.576
TASK C∗	FBK_C1	0.341	0.264
	ClearTK-2_EN	*n.a.*	0.309
	TIPSemB-F_ES	*n.a.*	0.416
TASK D∗	FBK_D1	0.731	0.736
	UTTime-1, 4_EN	*n.a.*	0.564

Table 7: Comparison with TempEval-3 systems.

Results for temporal expression detection, Task A, are above 0.80 in all languages. The results for normalization present a higher variability ranging from 0.709 for Italian up to 0.875 for Spanish. The lower results for Italian can be due to the fact that empty TIMEX3 tags were taken into account in the evaluation, while this was not done in TempEval-3. Still the difference between English and Italian is minor when compared to Spanish.

In Task B, event detection and normalization, system results are pretty similar for event detection but differ highly for the classification. This difference can be due mainly to the annotated data as all systems are comparable in terms of features used.

Finally, the analysis of Task D and C requires a *caveat*, namely that Task C, full temporal processing, has been simplified in Italian with respect to Task C in TempEval-3. Nevertheless, the results are very low, signaling that this task is very hard and that different approaches and solutions are to be envisaged.

9 Conclusion

This paper describes the EVENTI evaluation exercise within the EVALITA 2014 evaluation campaign. The task requires the participants to automatically annotate a raw text with temporal information. This involves the identification of temporal expressions, events and temporal relations. As for temporal relations, we have restricted the set of relations only to event-event and event-timex pairs in the same sentence.

The EVENTI evaluation exercise is the first end-to-end task on Temporal Processing for Ital-

ian and it is strictly linked to the TempEval-3 challenge. In particular, it adopts the same evaluation method thus aiming at facilitating comparison between systems developed in different languages. EVENTI is also the first evaluation on Temporal Processing of Historical Texts, organized to foster the collaboration between the NLP and the Digital Humanities communities.

Future work will aim at providing the full set of temporal relations without restrictions and possibly investigate temporal processing in specific applications or broader tasks (e.g. RTE and QA) both for Italian and from a multilingual perspective. The results obtained by the one end-to-end system participating in EVENTI show that there is still room for improvement in the identification and interpretation of temporal expressions, events, and temporal relations.

10 Acknowledgments

Our thanks to Nashaud UzZaman which has allowed us to re-use the evaluation script of TempEval-3 for the EVENTI Task, Giovanni Moretti for his assistence in trasforming the data to the CAT format, Anne-Lyse Minard for adapting the evaluation script.

References

G. Attardi, S. Dei Rossi, and M. Simi. 2010. The Tanl Pipeline. In *Proc. of LREC Workshop on WSPP*.

V. Bartalesi Lenzi and R. Sprugnoli. 2007. Evalita 2007: Description and Results of the TERN Task. *Intelligenza artificiale*, 2(IV):55–57.

V. Bartalesi Lenzi, G. Moretti, and R. Sprugnoli. 2012. CAT: the CELCT Annotation Tool. In *Proceedings of the Eighth International conference on Language Resources and Evaluation (LREC-12)*, pages 333–338.

S. Bethard. 2013. A Synchronous Context Free Grammar for Time Normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 821–826, Seattle, Washington, USA, October. Association for Computational Linguistics.

C. Bosco, F. DellOrletta, S. Montemagni, and M. Simi, editors. 2014. *Evaluation of Natural Language and Speech Tools for Italian*, volume 1. Pisa University Press.

T. Caselli, V.B. Lenzi, R. Sprugnoli, E. Pianta, and I. Prodanof. 2011a. Annotating events, temporal expressions and relations in italian: the it-TimeML experience for the Ita-TimeBank. In *Proceedings of the 5th Linguistic Annotation Workshop (LAW V)*, pages 143–151.

T. Caselli, H. Llorens, B. Navarro-Colorado, and E Saquete. 2011b. Data-driven approach using semantics for recognizing and classifying TimeML events in Italian. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 533–538.

T. Caselli. 2010. IT-TimeML: TimeML annotation scheme for Italian, version 1.3.1, technical report. Technical report, ILC-CNR, Pisa.

A. De Gasperi. 2006. Scritti e discorsi politici. In E. Tonezzer, M. Bigaran, and M. Guiotto, editors, *Scritti e discorsi politici*, volume 1. Il Mulino.

P. Mirza and S. Tonelli. 2014. Classifying Temporal Relations with Simple Features. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden.

E. Pianta, L. Bentivogli, and C. Girardi. 2002. MultiWordNet: developing an aligned multilingual database. In *First International Conference on Global WordNet*, Mysore, India.

J. Pustejovsky, J. Castao, R. Ingria, R. Saurì, R. Gaizauskas, A. Setzer, and G. Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*.

J. Pustejovsky, P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. 2003b. The TIMEBANK corpus. In *Corpus Linguistics 2003*.

L. Robaldo, T. Caselli, I. Russo, and M. Grella. 2011. From Italian Text to TimeML Document via Dependency Parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 6609 of *Lecture Notes in Computer Science*, pages 177–187. Springer Berlin / Heidelberg.

J. Strötgen and M. Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of SemEval 2010*, pages 321–324, Uppsala, Sweden, July. Association for Computational Linguistics.

J. Strötgen, A. Armiti, T. Van Canh, J. Zell, and M. Gertz. 2014. Time for More Languages: Temporal Tagging of Arabic, Italian, Spanish, and Vietnamese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):1–21.

N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky. 2013. SemEval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of SemEval-2013*, pages 1–9. Association for Computational Linguistics, Atlanta, Georgia, USA.

10.12871/clicit201426

Experiments in Identification of Italian Temporal Expressions

Giuseppe Attardi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
I-56127 Pisa, Italy
attardi@di.unipi.it

Luca Baronti
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3
I-56127 Pisa, Italy
lbaronti@gmail.com

Abstract

English. We describe our experiments in participating to the EValuation of Events aNd Temporal Information (EVENTI) task, for the EVALITA 2014 evaluation campaign. We used the HeidelTime tagger extended with a wrapper for the Tanl POS tagger and tokenizer of the Tanl suite. The rules for recognizing Italian temporal expressions were rewritten and extended after the submission, leading to a 10 point increase in F1 over the Italian rules in the HeidelTime distribution.

Italiano. *Nell'articolo descriviamo gli esperimenti svolti per la nostra partecipazione al task EValuation of Events aNd Temporal Information (EVENTI), nel'ambito della campagna di valutaionze EVALITA 2014. Per il riconoscimento e normalizzazione delle espresioni temporali abbiamo utilizzato il tagger HeidelTime, estendendolo con un wrapper per poter utilizzare il POS tagger e il tokenizer della suite di NLP Tanl. Le regole per il riconoscimento delle espressioni temporali in italiano sono state riscritte ed estese, dopo la sottomissione, ottenendo un miglioramento di 10 punti di F1 rispetto alle regole presenti nella distribuzione di HeidelTime.*

1 Introduction

The shared task EVENTI at Evalita 2014, required to recognize temporal expressions within a corpus of Italian text documents and to normalize them according to the It-TimeML TIMEX3 specifications.

Training and test data are distributed in the CAT (*Content Annotation Tool*) (Bartalesi Lenzi et al., 2012) labelled format. This is an XMLbased standoff format where different annotation layers are stored in separate document sections and are related to each other and to source data through pointers.

2 Approach

We chose to use an available temporal tagger and to adapt it for the task. HeidelTime (2014) is a cross-domain temporal tagger developed at the Database Systems Research Group at Heidelberg University (Strötgen and Gertz, 2013). For detecting temporal expressions, HeidelTime uses a set of rules based on regular expressions and conditions on the POS tags of words matched by those expressions. The rules also contain normalization directives for producing the TIMEX3 notation.

HeidelTime provides a plugin architecture, relying on external tools for performing tokenization and POS tagging. The current distribution provides wrappers for TreeTagger (Schmid, 1994), Stanford POSTagger and JVnTextPro.

The standalone version of HeidelTime requires a plain text as input and returns a TimeML (Pustejovsky et al., 2003) document containing the original text with the temporal expressions enclosed within a TIMEX3 element.

HeidelTime is based on the UIMA architecture, that orchestrates the processing of data among CAS processors, passing CAS objects from one stage to the next forming a pipeline. Typically the HeidelTime pipeline consists in three stages: tokenization, POS tagging and sen-

tence annotation. The first two stages are delegated to wrappers for external tools, the third one is dealt by HeidelTime itself. Those tools that provide a UIMA interface are called directly in memory; the others are invoked through wrappers that pass them as input a plain text file and collect the annotations to be added to the CAS from their output. This is the case for TreeTagger.

In our case, we wished to use the tools from Tanl (Text Analytics in Natural Language) (Attardi, Dei Rossi and Simi, 2010) a suite of statistical machine learning tools for text analytics based on the software architecture paradigm of data pipelines. Differently from UIMA, where each stage must process a whole document before it can be handled to the next stage, in a data pipeline processing occurs on demand and each stage pulls data as needed from its preceding stage. The granularity of the units of data requested at each stage depends on the requirements of that stage and can vary from a single line of text, a single token or a single sentence.

The Tanl POS tagger (Attardi et al., 2010) is similar to the one that achieved the best results (Attardi et al., 2009) in the task of POS classification at Evalita 2009.

3 Format Conversion

The training corpus is provided in CAT format where text is represented as an ordered list of tokens. The temporal expression information is present in TIMEX3 elements within the Markables element after the tokens. A temporal event in the text is represented by a TIMEX3 element with attributes representing its type and value, and with children elements containing numeric references to the tokens involved.

A special TIMEX3 element with no children is used to store the publication time information[1], useful for the tagger to correctly compute the absolute time for relative[2] temporal expressions like "ieri" or "lo scorso giugno".

A scorer script is provided by the organizers for evaluating the accuracy of a system output. The scorer works with two sets of CAT files, typically the gold annotated reference set and the system output.

We process each document through the following steps:

1. extract the publication/creation date from the document;
2. convert the corpus document to plain text or use the supplied text version of it;
3. invoke HeidelTime supplying both the plain text file and the publication date as parameters;
4. convert the HeidelTime output into CAT format.

Each step, except the 3rd, is performed by a suitable Python script. The whole process is driven by a custom Makefile, in order to automate the process of carrying out or of repeating the experiments.

4 Results

Before the submission deadline we didn't have time to perform any fine tuning of the HeidelTime rules for Italian. Instead, we focused on integrating the Tanl tagger into the HeidelTime pipeline, and to test its out-of-the-box performance. Hence, we didn't exploit the training corpus for tuning or correcting the rules for Italian, and we used a basic model for the Tanl tagger.

We submitted two runs: Unipi_Tanl and Unipi _TreeTagger. Unipi_TreeTagger is a baseline run produced using HeidelTime in its default configuration for Italian, using TreeTagger and the supplied Italian rules. Unipi_Tanl was an attempt to use the tools from Tanl (the Tanl Tokenizer and the Tanl POS tagger) adapting the rules for using the Tanl POS tagset. Unfortunately, as we discovered later delving into the code of HeidelTime, the rules for matching POS tags were written using regular expressions, which turned out not to be supported in the current version of HeidelTime.

This explains why the official scores in Table 1 show no significant difference between the two runs. We corrected this problem after the submission and rewrote the rules for Italian as described in the following section, achieving significant improvements.

POS Tagger	F1 (strict)	F1 (relaxed)
Best	**0.821**	**0.893**
Unipi_Tanl	0.659	0.771
Unipi_TreeTagger	0.662	0.768

Table 1. Results in Task A.

[1] sometimes different from "creation time".

[2] As opposed to an absolute temporal expression like "23 dicembre 1934" which can be correctly tagged without reference to the publication time.

5 Wrapper for TanlTagger

Proper use of the Tanl POS Tagger with HeidelTime requires adding a wrapper for it to the HeidelTime sources.

We wrote a Java class HeidelTimeWrapper, which invokes the Tanl Tokenizer and the Tanl POS Tagger as subprocesses. An even better solution would be to build a CAS processor interface for these tools.

A few changes were required also to the code of HeidelTime itself. In particular for dealing with POS_CONSTRAINT rules, which apply only if the expression belongs to a specified POS class, the original code performed a simple string match between the requested POS and the one in the data. However, the POS tags produced by the Tanl Tagger are more refined than those of TreeTagger and include morphology information. One rule for example involves checking whether a word is a plural noun, but since nouns have both number and gender, it is required to check for either `Smp` (noun, male, plural) or `Sfp` (noun, female, plural). Hence we modified the code to allow specifying constraints by means of regular expressions, so that one could just write `S.p`.

We also had to fix a bug in the code that added an extra empty line and skipped the final newline in the file passed to the tokenizer, which caused misalignments in tokens.

Both these changes were reported to the maintainers of the package and will be included in later releases of HeidelTime.

We also stumbled upon another bug in the rule matching code of HeidelTime: when a pattern contains an alternative like this "`(%reUnit|%reUnitTime)`", where the first alternative is a substring of the second, the second one is discarded.

Furthermore, we discovered another unexplained idiosyncrasy in some pattern behavior that was solved by adding a "\b" in front of them.

6 Error Analysis

In order to analyze the tagger errors, we developed a diff script that compares two CAT documents and lists their differences, i.e. each timex3 present in one and missing from the other and vice versa. The script also signals expressions that are tagged with a different type/value.

On the development set our system achieves these values of accuracy:

	Precision	Recall	F1
strict	0.800	0.809	0.805
relaxed	0.884	0.894	0.889

Table 2. Development results.

The absolute values of the True Positives, False Positives and False Negatives on the training corpus are the following:

	TP	FP	FN
strict	633	149	158
relaxed	699	83	92

Table 3. True and False Positives on the training set.

We investigated the causes of the large number of False Positives. Inspecting the output of our comparison script shows that the errors can be classified into the following types:

- adverbs like `presto`/`subito` or adjectives like `passati`/`futuro` that are excluded in the guidelines
- person ages (`51 anni`)
- double digit numbers (`83`, `86`)
- minor differences, e.g. in the extent of the expression or different time value
- a few legitimate temporal expressions (`una settimana fa`, `mese di settembre`, `notte prima`, `alle 23`, `lunedì`, `prossimo anno`, `ultimo trimestre`).

Further tuning the HeidelTime rules might hence help reducing these errors.

The situation with False Negatives is more complicated. Here is a small sample:

```
91
l'anno
86
data
90
un minimo di cinque
un massimo di quindici anni
l'81
quattro ore tutte le mattine
Verso le 9.3
qualche mese a questa parte
in futuro
ora in avanti
solo mese di settembre
ventiquattr'ore dopo
mese tradizionalmente "caldo"
meno di due anni
oltre un anno
```

A few of these are actually ambiguous (`91`, `86`, `90`, `data`) and would require deeper analysis to

be recognized as years; some are due to problems in HeidelTime rule matching (`l'81`, `qualche mese a questa parte`, `oltre un anno`); others have patterns that could actually be dealt by additional specific rules.

Using the diff script we were able to address several misclassification problems, improving the HeidelTime rule system for Italian. The rules included for Italian in the standard distribution of HeidelTime contained a lot of errors. Many seem due to the fact that the rules appear to be incomplete translations from the Spanish version, as shown in this rule:

```
[Pp]rimera met(àa')
```

which should read instead

```
[Pp]rima met(àa')
```

In order to improve the accuracy we almost completely rewrote the rules for Italian and devoted some effort also in making them more modular, avoiding idiosyncrasies and repetitions.

7 After Submission Results

After revising the Italian rule set, we performed a run on the test set, using the new wrapper for the TanlTagger, achieving a significant accuracy improvement, as reported in Table 4.

POS Tagger	F1 (strict)	F1 (relaxed)
Best	**0.821**	**0.893**
Unipi_Tanl	0.723	0.871

Table 4. After submission results.

8 Conclusions

We explored a rule based approach to identification and normalization of temporal expressions in Italian documents.

We chose to use the HeidelTime kit, which allows developing resources for different languages using a suitable rule syntax.

HeidelTime has already been used in other challenges achieving top results on English documents at the TempEval-2 challenge in 2010.

The rules for Italian provided in the distribution turned out to be fairly poor. By rewriting and extending them we were able to achieve a significant 10 point improvement in F1 relaxed accuracy, reaching a score not far from the best. It should be possible to close the gap with some additional effort. We were slowed down in doing so by stumbling upon some problems in the rule matching algorithm of HeidelTime version 1.7, that are due to be fixed in release 1.8.

In order to better deal with Italian documents, we wrote a wrapper for the Tanl POS tagger, which is reported as one of the best for Italian. The use of POS tags is still fairly limited though: for instance it is used to distinguish whether a four digit number is not a year, by the fact that it is followed by a plural noun. More extensive of rules involving POS constraints might help eliminate some false positives.

An interesting development would be to apply more sophisticated analysis tools, for instance a parser. Compositional meaning representations of temporal expressions could be reconstructed from phrases that contain temporal clues and machine learning could be applied to learn their interpretation as in (Angeli and Uszkoreit, 2013) or (Leey et al., 2014).

References

Gabor Angeli and Jakob Uszkoreit. 2013. Language-Independent Discriminative Parsing of Temporal Expressions. *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics* (ACL 2013).

Giuseppe Attardi and Maria Simi. 2009. Overview of the EVALITA 2009 Part-of-Speech Tagging Task. *Proc. of Workshop Evalita 2009*.

Giuseppe Attardi, Stefano Dei Rossi and Maria Simi. 2010. The Tanl Pipeline. *Proc. of LREC 2010 Workshop on WSPP*, Malta.

Valentina Bartalesi Lenzi, Giovanni Moretti and Rachele Sprugnoli. 2012. CAT: the CELCT Annotation Tool. In *Proceedings of LREC 2012*, 333–338.

HeidelTime. 2014. Version 1.7. Retrieved from http://code.google.com/p/heideltime/

Kenton Leey, Yoav Artziy, Jesse Dodgez, and Luke Zettlemoyer. 2014. Context-dependent Semantic Parsing for Time Expressions.

James Pustejovsky, José M. Castano, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, Dragomir R. Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. *New Directions in Question Answering*, 28–34.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, number 1, 269–298. Springer.

10.12871/clicit201427

HeidelTime at EVENTI: Tuning Italian Resources and Addressing TimeML's Empty Tags

Giulio Manfredi and **Jannik Strötgen** and **Julian Zell** and **Michael Gertz**
Institute of Computer Science, Heidelberg University, 69120 Heidelberg, Germany
manfredi@stud.uni-heidelberg.de,
{stroetgen,zell,gertz}@informatik.uni-heidelberg.de

Abstract

English. In this paper, we describe our participation in the EVENTI task. We addressed subtask A, the extraction and normalization of temporal expressions in Italian texts, by adapting our existing multilingual temporal tagger HeidelTime. In addition to improving its ability to handle Italian texts, we added further functionality to support empty tags. Based on the main evaluation criterion, HeidelTime ranked first among the participating systems. The new HeidelTime version is publicly available.[1]

Italiano. *In questo articolo descriviamo la nostra partecipazione al task EVENTI. Ci siamo dedicati al sottotask A, cioè l'estrazione e normalizzazione di espressioni temporali all'interno di testi in lingua italiana, e a questo scopo abbiamo adattato il nostro temporal tagger multilingue, HeidelTime. Oltre a migliorare le sue capacità di elaborare testi in italiano, abbiamo aggiunto nuove funzionalità per supportare i tag vuoti. In base al principale criterio di valutazione, HeidelTime è risultato primo rispetto agli altri sistemi che hanno partecipato al task. La nuova versione di HeidelTime è disponibile pubblicamente.*[1]

1 Introduction

EVENTI (EValuation of Events aNd Temporal Information) is a task of EVALITA 2014, an initiative aimed at the evaluation of NLP tools for Italian.[2] It comprises four subtasks: the extraction and normalization of temporal expressions, i.e., temporal tagging (A), the extraction of events (B), and the annotation of temporal relations (C, D).

Together, they form the task of temporal annotation, which is helpful in many natural language processing and understanding applications such as question answering and summarization. But even the temporal tagging subtask itself is valuable for many applications, e.g., in information retrieval (Alonso et al., 2011; Campos et al., 2014).

In this paper, we describe our efforts to address the temporal tagging subtask of EVENTI, for which we extended and improved our temporal tagger HeidelTime (Strötgen and Gertz, 2013). In addition to earlier approaches to Italian temporal tagging (e.g., Negri 2007) and to manually annotated Italian corpora (Magnini et al., 2006), Italian was also one of six languages offered at TempEval-2 (Verhagen et al., 2010). However, participants only addressed English and Spanish, and we also added Italian to HeidelTime only more recently (Strötgen et al., 2014). While Italian had thus already been implemented in HeidelTime, there was room for improvement in the context of the EVENTI challenge as will be detailed in this paper. As reference point for our work, the EVENTI task guidelines (Caselli et al., 2014) and the Ita-TimeBank corpus (Caselli et al., 2011) – newly released as training data – were used.

The rest of the paper is structured as follows. After an overview of HeidelTime's architecture and challenges that needed to be addressed, our adaptations to HeidelTime are detailed in Section 3. In Section 4, evaluation results are reported and compared to those of HeidelTime's previous version and the systems of the other participants.

2 Starting Point & Challenges

In this section, we first describe HeidelTime's architecture and then explain the challenges that had to be addressed although HeidelTime already supported Italian temporal tagging.

[1] http://code.google.com/p/heideltime/
[2] http://www.evalita.it/2014

2.1 HeidelTime's Architecture

HeidelTime is a rule-based, multilingual, and cross-domain temporal tagger initially developed for English in the context of TempEval-2 (Strötgen and Gertz, 2010). It is based on the Unstructured Information Management Architecture[3] (UIMA), which allows to easily combine different modules because all rely on the same data structure, called *Common Analysis Structure* (CAS).

In a UIMA pipeline for temporal tagging with HeidelTime, input documents are first read by a *collection reader*, which initializes a CAS object for each document. The subsequent tasks are sentence splitting, tokenization, and part-of-speech tagging before HeidelTime itself is called. The TreeTagger for Italian linguistic preprocessing (Schmid, 1994), and HeidelTime are employed as *analysis engines*. Eventually, the output is created by a *CAS consumer*, which writes the text and its annotations to a database or file.

An important characteristic of HeidelTime's architecture is the strict separation of source code and language dependent resources. This allows adding new languages and improving already implemented ones without affecting the functionality of the system itself and without requiring a deep knowledge of its mechanisms. Several languages were thus integrated by different research groups: German (Strötgen and Gertz, 2011), Dutch (van de Camp and Christiansen, 2012), Spanish (Strötgen et al., 2013), French (Moriceau and Tannier, 2014), Italian, Arabic, Vietnamese (Strötgen et al., 2014), Chinese (Li et al., 2014), Russian, and Croatian (Skukan et al., 2014).

HeidelTime's language resources are of three types: patterns, normalizations, and rules. There is one rule file for each possible value of the TIMEX3 *type* attribute (`DATE`, `TIME`, `DURATION` and `SET`), and each rule has three mandatory fields: `RULENAME`, `EXTRACTION` and `NORM_VALUE`. The `EXTRACTION` field is a regular expression that also contains references to the patterns, which are themselves sets of regular expressions. The field `NORM_VALUE` uses the normalization resources to translate the patterns into a standard format and to normalize extracted temporal expressions according to the TimeML specifications (Pustejovsky et al., 2003).[4]

[3] `http://uima.apache.org/`

[4] For further details about HeidelTime's rule syntax, we refer to (Strötgen and Gertz, 2013).

2.2 Challenges for EVENTI Participation

HeidelTime's initial resources for Italian were developed on the Italian TempEval-2 training data (Strötgen et al., 2014), although the TempEval-2 corpus developers stated that the non-English "annotations are a bit experimental" (Verhagen, 2011). Thus, using now more sophisticated guidelines and training data, several adaptations were required. With regard to language-dependent resources, most work consisted of extending patterns, adding rules, and improving existing ones.

Furthermore, a main challenge was that in the EVENTI data, empty TIMEX3 tags – which represent implicit temporal information – are considered. Although such empty tags are also defined in the original TimeML annotation specifications,[5] they have hardly been considered so far, neither in manually annotated corpora nor in research competitions nor by temporal taggers. They were also not created by HeidelTime so far, and were thus a feature that needed to be implemented.

Finally, the particular format of the EVENTI training and test data required specific tools to read the documents and output HeidelTime's annotations in the required format as described below.

3 HeidelTime Adaptations

Our efforts can be split into three parts: developing UIMA components, extending HeidelTime, and improving HeidelTime's Italian resources.

3.1 UIMA Components for EVENTI Data

The EVENTI training and test data consist of Ita-TimeBank documents (news articles). Each document is provided as an XML file containing sentence and token annotations. In the training data, TIMEX3 tags are additionally provided.

To handle this format at the input and output stages, we wrote a collection reader and a CAS consumer. These are also part of the new HeidelTime-kit, which allows to easily reproduce our evaluation results on the EVENTI data.

3.2 Empty TIMEX3 Tags

The main feature we needed to add, though, was the creation of empty tags. These are part of the It-TimeML specifications but were not present in previous temporal tagging corpora and competitions. Empty tags are TIMEX3 tags that do not

[5] `http://timeml.org/site/publications/timeMLdocs/timeml_1.2.1.html`.

contain any tokens and should be created whenever a temporal expression can be inferred from already existing text-consuming TIMEX3 tags. Two cases are implicit begin and end points of temporal expressions of type DURATION, e.g., *un mese fa* (a month ago), and implicit durations which can be deduced from two TIMEX3 tags of type DATE, e.g., *dal 2010 al 2014* (from 2010 to 2014). We refer to the former as *anchored durations* and to the latter as *range expressions*.[6]

To handle anchored durations, we modified HeidelTime's rule syntax by adding an additional field, called EMPTY_VALUE. It is syntactically similar to NORM_VALUE and contains an offset to a reference time. This offset, combined with the value returned by NORM_VALUE, is then used by HeidelTime to compute a normalized date. Note that this EMPTY_VALUE extension is language-independent and had to be realized by modifying HeidelTime's source code.

To extract range expressions, the UIMA HeidelTime kit already contained an analysis engine called Interval Tagger, which creates TIMEX3 independent temporal annotations. So far, however, only English interval rules were available, and not TIMEX3 duration values but start and end time points of range expressions were determined. In addition to writing Italian rules, we thus added the ability to calculate the difference between the two DATE expressions, i.e., duration values for range expressions, as defined in the specifications.

In both cases, the computed values are included as additional, HeidelTime-internal attributes to text-consuming TIMEX3 annotations. Our EVENTI CAS consumer reads out these attributes to print empty TIMEX3 tags with the respective *value* information. Furthermore, it adds to each empty tag a reference to the TIMEX3 tag(s) that triggered it.

3.3 Tuning Italian Resources

Despite the efforts required to implement the empty tag feature, most time was spent on extending the existing Italian resources. This was done by carefully applying the guidelines provided by the EVENTI task organizers. While modifying normalization information of existing patterns was rather simple, quite a lot of work was needed to improve the performance in the extraction phase.

[6] A third empty tag type is described as further challenge in Section 4 since we have not yet addressed it.

Since HeidelTime is a rule-based system that makes use of regular expressions, new patterns were added to extract expressions which had not been considered before and, as a consequence, to improve the recall of the system. While doing this, we tried to write the rules as general as possible without producing many false positives. In Italian, however, there are several expressions that can be ambiguous and therefore require context knowledge to be correctly interpreted. Obviously, this is somewhat limited by the abilities of a rule-based system and thus particularly challenging.

An example is the adverb *allora*, which, depending on the context, can mean "at that time" or "therefore". Our system only identifies the temporal meaning if it can be inferred from neighboring words, as in *già allora* (already at that time).

Some of the patterns that were added are those representing sets of months or years, e.g., *bimestre* (two months) and *lustro* (five years), and specific post-modifiers that affect the normalization of an expression, e.g., *esaminato*, *in discussione* and *di che trattasi*, all referring to the period of time that is being dealt with.

4 EVENTI Evaluation

The extraction quality of all participating systems and of all runs of each system is evaluated using precision, recall, and F1-score for strict and relaxed matches. To evaluate normalization abilities, the accuracy of the *type* and *value* attributes are multiplied by the F1-score for strict matches in order to normalize it. The resulting *value F1* measure is used as main evaluation criterion.

Table 1 shows official results of all participating teams. We submitted three runs: HeidelTime 1.7 (publicly available before EVENTI), HeidelTime 1.8 (comprising all adaptations described above), and version 1.8 without the empty tags feature. With regard to this aspect, the measures show only small differences, mainly because empty tags are rare compared to other tags. Although precision is slightly higher when ignoring empty tags, recall, F1-score, and normalization quality increase significantly when taking them into account.

Most important, however, is the massive improvement of HeidelTime 1.8 over 1.7 with respect to extraction and normalization quality.

The extraction quality of the system of team B is similar to HeidelTime 1.8. Its F1-score is slightly higher for strict but lower for relaxed matches.

	relaxed match			strict match			normalization	
	P	R	F1	P	R	F1	type F1	**value F1**
HT 1.7	92.1	67.6	78.0	78.2	57.4	66.2	64.3	57.1
HT 1.8 (no ET)	94.0	82.4	87.8	86.1	75.5	80.4	77.5	69.0
HT 1.8	93.5	85.4	*89.3*	86.0	78.5	82.1	79.2	**70.9**
Team B-1	93.6	84.1	88.6	87.3	78.5	*82.7*	80.0	66.5
Team C-1	92.9	65.4	76.8	80.2	56.4	66.2	64.3	56.6
Team C-2	92.2	66.2	77.1	78.8	56.6	65.9	64.0	56.3

Table 1: EVENTI evaluation results on test data.

With respect to the normalization quality, HeidelTime outperforms team B by 4.4 and team C by 14.3 percentage points (value F1).

Finally, comparing HeidelTime's performance on the test set and the FBK and ILC training sets reveals some differences. While value F1 is only slightly higher on the FBK set (73.5), it is much higher on the ILC set (84.2) – mainly due to many rather difficult expressions in the FBK set.

4.1 Error Analysis

In general, four error types can be distinguished: false positives, false negatives, partial matches, and incorrect normalizations. Although the main evaluation criterion combines correct value normalization with strict matching, in our opinion, value F1 with relaxed matching is even more meaningful (HeidelTime 1.8: 74.7). Expressions that are only partial matches but correctly normalized are often equally valuable as correctly normalized strict matches for any NLP or IR tasks relying on temporal taggers.

Considering relaxed matching, only 37 false positives are extracted by HeidelTime, and of 624 gold expressions, 533 are retrieved with either strict or relaxed matching. 446 of them are additionally normalized correctly.

Simple examples of partial matches with correct value normalization are expressions such as *un lasso di tempo di 14 giorni* (a lapse of time of 14 days), where HeidelTime extracts only *14 giorni*, but the normalization is correct.

A further issue occurs if two tags are created instead of one. Instead of *ieri verso le 11* (yesterday around 11), HeidelTime extracts *ieri* and *verso le 11* separately. Nonetheless, the value of *verso le 11* is the same as the gold annotation. Considering strict matching, such mistakes generate two false positives and one false negative.

A reason for incorrect normalizations is that several `DATE` expressions have a value of `XXXX-XX-XX` in the gold standard. HeidelTime, however, tries to resolve extracted `DATE` expressions instead of leaving them unspecified. Another reason is the occurrence of `TIME` values that contain a time without date in the gold standard. However, it is often difficult to decide if a `TIME` expression refers to a specific day or if it is used generically. HeidelTime usually assigns values to `TIME` expressions with specified day information. Furthermore, its strategy to select the previously mentioned day as reference day is sometimes incorrect. Often, however, this strategy works fine as in the example above where *ieri* is selected as reference time for the expression *verso le 11*.

4.2 Open Challenges

What needs to be addressed in the future is a third category of expressions that generate empty tags, namely *framed durations*. These are durations located in a specific time frame and for which a begin and an end point can be inferred. An example is *i primi 6 mesi dell'anno* (the first 6 months of the year), where, in addition to a `DURATION` (*i primi 6 mesi*) and a `DATE` (*anno*), two additional `DATE` expressions can be deduced, referring to the first and sixth month of the year in question. Thus, two empty tags with values pointing to January and June of the respective year should be created.

A further example of an ambiguity issue in addition to the ones described in Section 3.3, are expressions referring to ages which are often ambiguous in Italian. For instance, the Italian expression *26 anni* can mean "26 year old" or "26 years" – but only in the latter case it should be annotated.

Finally, the creation of empty tags has been developed specifically for the EVENTI task, so that it is currently only available for Italian. However, the expansion to the other languages supported by HeidelTime should not be time consuming because it merely requires an adaptation of the rules.

5 Summary

In this paper, we described our participation in the temporal tagging task of EVENTI 2014. By extending HeidelTime to cover TimeML's empty TIMEX3 tags and by tuning HeidelTime's Italian resources based on high quality specifications and training data, we significantly improved HeidelTime's tagging quality for Italian. We outperformed the other participants' systems by at least 4.4 percentage points for correct extraction and normalization (value F1).

References

Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. 2011. Temporal Information Retrieval: Challenges and Opportunities. In *Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW 2011)*, pages 1–8.

Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. 2014. Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys*, 47(2):15:1–15:41.

Tommaso Caselli, Valentina Bartalesi Lenzi, Rachele Sprugnoli, Emanuele Pianta, and Irina Prodanof. 2011. Annotating Events, Temporal Expressions and Relations in Italian: the It-TimeML Experience for the Ita-TimeBank. In *Proceedings of the 5th Linguistic Annotation Workshop (LAW 2011)*, pages 143–151.

Tomasso Caselli, Rachele Sprugnoli, Manuela Speranza, and Monica Monachini. 2014. EVENTI: EValuation of Events aNd Temporal Information Task Guidelines for Participants v 1.0. Technical report, TrentoRISE, FBK, University of Trento, and ILC-CNR.

Hui Li, Jannik Strötgen, Julian Zell, and Michael Gertz. 2014. Chinese Temporal Tagging with HeidelTime. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 133–137.

Bernardo Magnini, Emanuele Pianta, Christian Girardi, Matteo Negri, Lorenza Romano, Manuela Speranza, Valentina Bartalesi Lenzi, and Rachele Sprugnoli. 2006. I-CAB: the Italian Content Annotation Bank. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 963–968.

Véronique Moriceau and Xavier Tannier. 2014. French Resources for Extraction and Normalization of Temporal Expressions with HeidelTime. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 3239–3243.

Matteo Negri. 2007. Dealing with Italian Temporal Expressions: The ITA-CHRONOS System. In *Proceedings of Evalita 2007*.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *New Directions in Question Answering*, pages 28–34.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Luka Skukan, Goran Glavaš, and Jan Šnajder. 2014. HEIDELTIME.HR: Extracting and Normalizing Temporal Expressions in Croatian. In *Proceedings of the 9th Slovenian Language Technologies Conferences (IS-LT 2014)*, pages 99–103.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pages 321–324.

Jannik Strötgen and Michael Gertz. 2011. WikiWarsDE: A German Corpus of Narratives Annotated with Temporal Expressions. In *Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2011)*, pages 129–134.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298.

Jannik Strötgen, Julian Zell, and Michael Gertz. 2013. HeidelTime: Tuning English and Developing Spanish Resources for TempEval-3. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 15–19.

Jannik Strötgen, Ayser Armiti, Tran Van Canh, Julian Zell, and Michael Gertz. 2014. Time for More Languages: Temporal Tagging of Arabic, Italian, Spanish, and Vietnamese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):1–21.

Matje van de Camp and Henning Christiansen. 2012. Resolving Relative Time Expressions in Dutch Text with Constraint Handling Rules. In *Constraint Solving and Language Processing – 7th International Workshop (CSLP 2012)*, pages 166–177.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pages 57–62.

Marc Verhagen. 2011. TempEval2 Data – Release Notes. Technical report, Brandeis University.

10.12871/clicit201428

FBK-HLT-time: a complete Italian Temporal Processing system for EVENTI-Evalita 2014

Paramita Mirza
FBK, Trento, Italy
University of Trento
paramita@fbk.eu

Anne-Lyse Minard
FBK, Trento, Italy
minard@fbk.eu

Abstract

English. In this paper we present an end-to-end system for temporal processing of Italian texts based on a machine learning approach, specifically supervised classification. The system participated in all subtasks of the EVENTI task at Evalita 2014 (identification of time expressions, events, and temporal relations), including the pilot task on historical texts.

Italiano. *In questo articolo presentiamo un sistema end-to-end per l'analisi temporale su testi in italiano basato su algoritmi di apprendimento automatico (classificazione supervisionata). Il sistema ha partecipato a tutti i sottotask di EVENTI a Evalita 2014 (individuazione di espressioni di tempo, eventi e relazioni temporali), incluso il task pilota relativo a testi storici.*

1 Introduction

Research on temporal processing has been gaining a lot of attention from the NLP community in the recent years. The goal is to automatically extract events and temporal information from texts in natural language. The most recent shared task, TempEval-3 (UzZaman et al., 2013), focused on these goals. However, even though TempEval-3 organizers also released annotated data in Spanish, English is still given the most attention.

EVENTI[1], one of the new tasks of Evalita 2014[2], is established to promote research in temporal processing for Italian texts. Currently, even though there exist some independent modules for temporal expression extraction (e.g. HeidelTime (Strötgen et al., 2014)) and event extraction (e.g. Caselli et al. (2011)), there is no complete system for temporal processing for Italian. The main EVENTI task is composed of 4 subtasks for time expression recognition and normalization, event detection and classification and temporal relation extraction from newspaper articles. A pilot task on temporal processing of historical texts was also proposed. Our system participated in both tasks.

In this paper, we summarize our attempts and approaches in building a complete extraction system for temporal expressions, events, and temporal relations, which participates in the EVENTI challenge.

2 End-to-end system

We developed an end-to-end system to participate in the EVENTI challenge. It combines three subsystems: *(i)* time expression (timex) recognizer and normalizer, *(ii)* event extraction and *(iii)* temporal relation identification and classification. The subsystems used have been first developed for English as part of the NewsReader project[3] and then adapted to Italian. In order to adapt and test them for Italian, we used the training data released by the task organizers and split them into development and test data (in 80%/20% proportion).

The timex normalizer includes an adaptation of TimeNorm developed by Bethard (2013) for English, based on synchronous context free grammars. The other subsystems are based on machine learning and use Support Vector Machines algorithm. All subtasks, except the timex normalization subtask, are treated as classification problems. The feature sets used for building the classification models share a common ground, including morphological, syntactical and contextual features. The best combination of features and pre- and post-processing steps have been selected on the basis of experiments performed on the development data. The

[1] https://sites.google.com/site/eventievalita2014/

[2] http://www.evalita.it/2014

[3] http://www.newsreader-project.eu/

models used in the final system runs for the challenge have been trained on the whole training data.

3 Data and Tools

3.1 Data

The training data, the EVENTI corpus, is a simplified annotated version of the Ita-TimeBank released by the task organizers for developing purpose, containing 274 documents and around 112,385 tokens.

3.2 Tools

- **TextPro**[4] (Pianta et al., 2008), a suite of NLP tools for processing English and Italian texts. Among the modules we use: lemmatizer, morphological analyzer, part-of-speech tagger, chunker, named entity tagger and dependency parser.
- **YamCha**[5], a text chunker which uses SVMs algorithm. YamCha supports the dynamic features that are decided dynamically during the classification. It also supports multi-class classification using either *one-vs-rest* or *one-vs-one* strategies.
- **Snowball Italian stemmer**[6], a library for getting the stem form of a word.

3.3 Resources

- **MultiWordNet**[7], a multilingual lexical database containing WordNet aligned with the Italian WordNet. We extracted a list of words and their domains (e.g. *ricerca* [research] is associated to the domain *factotum*).
- **derIvaTario lexicon**[8], an annotated lexicon of about 11,000 Italian derivatives.
- **Lists of temporal signals** extracted from the training corpus. Mirza and Tonelli (2014) shows that the system performance benefits from distinguishing event-related signals (e.g. *mentre [while]*) from timex-related signals (e.g. *tra [within]*), therefore we split the list of signals into two separate lists.

4 Timex Extraction System

4.1 Timex Extent and Type Identification

The task of recognizing the extent of a timex, as well as determining the timex type (i.e. DATE, TIME, DURATION and SET), can be taken as a text chunking task. Since the extent of timex can be expressed by multi-token expressions, we employ the IOB2 tagging[9] to annotate the data. In the end, the classifier has to classify a token into 9 classes: B-DATE, I-DATE, B-TIME, I-TIME, B-DURATION, I-DURATION, B-SET, I-SET and O (for other).

The classifier is built using YamCha. One-vs-rest strategy for multi-class classification is used. The following features are defined to characterize a token:

- Token's text, lemma, part-of-speech (PoS) tags, flat constituent (noun phrase or verbal phrase), and the entity's type if the token is part of a named entity;
- Whether a token matches regular expression patterns for unit (e.g. *secondo [second]*), part of a day, name of days, name of months, name of seasons, ordinal and cardinal numbers, year (e.g. *'80*, *2014*), time, duration (e.g. *1h3'*, *50"*), temporal adverbs, names (e.g. *natale [Christmas]*), set (e.g. *mensile [monthly]*), or temporal signal as defined in TimeML;
- All of the above features for the preceding two and following two tokens, except the token's text;
- The preceding two labels tagged by the classifier.

4.2 Timex Value Normalization

For timex normalization, we decided to extend TimeNorm[10] (Bethard, 2013) to cover Italian time expressions. For English, it is shown to be the best performing system for most evaluation corpora compared with other systems such as HeidelTime (Strötgen et al., 2013) and TIMEN (Llorens et al., 2012).

We translated and modified some of the existing English grammar into Italian. Apart from the grammar, we modified the TimeNorm code in order to support Italian language specificity: normalization of accented letters, unification of articles and articulated prepositions, and handling the token splitting for Italian numbers that are concatenated (e.g. *duemilaquattordici [two thousand fourteen]*).

TimeNorm parses time expressions, and given an anchor time returns all possible normalizations following TimeML specifications. The anchor time

[4] http://textpro.fbk.eu/

[5] http://chasen.org/~taku/software/yamcha/

[6] http://snowball.tartarus.org/algorithms/italian/stemmer.html

[7] http://multiwordnet.fbk.eu

[8] http://derivatario.sns.it/

[9] IOB2 tagging format is a common tagging format for text chunking. The B- prefix is used to tag the beginning of a chunk, and the I- prefix indicates the tags inside a chunk. The label O indicates that a token belongs to no chunk.

[10] http://github.com/bethard/timenorm

passed to TimeNorm is always assumed to be the document creation time.

We have added post-process rules in order to select one of the returned values. The system chooses the value format that is most consistent with the timex type. For example if the timex is of type DURATION, the system selects the value starting with `P` (for Period of time).

After evaluating TimeNorm on the training data, we have added some pre-processing and post-processing steps in order to improve the performance of the system. The pre-processing rules treat time expressions composed by only one or two digits, and append either a unit or a name of month, which is inferred from a nearby timex or from the document creation time (e.g. *Siamo partiti il* $\underline{7_{timex}}$ *[We left (on) the 7]* (`DCT=2014-09-23 tid="t0"`) → $\underline{\text{7 settembre}_{timex}}$ [September 7]). We noticed that the TimeNorm grammar does not support the normalization of the *semester* or *half-year* unit (e.g. *il primo semestre [the first semester]*). In order to cope with this issue, we have developed some post-processing rules. Despite that, some expressions cannot be normalized because they are too complex, e.g. *'ultimo trimestre dell'anno precedente' [last quarter of the previous year]*.

4.3 Empty Timex Identification

The EVENTI annotation guidelines specifies the creation of empty TIMEX3 tags whenever a temporal expression can be inferred from a text-consuming one. For example, for the expression "*un mese fa [one month ago]*" two TIMEX3 tags are annotated: *(i)* one of type DURATION that strictly corresponds to the duration of one month (`P1M`) and *(ii)* one of type DATE that is not text consuming, referring to the date of one month ago.

As these timex are not text consuming they cannot be discovered by the text chunking approach. We performed the recognition of the empty timex using some simple post-processing rules and the timex normalization module.

5 Event Extraction System

Event detection is taken as a text chunking task, in which tokens have to be classified in two classes: EVENT (i.e. the token is included in an event extent) or O (for other). Then events are classified into one of the 7 TimeML classes: OCCURRENCE, STATE, I_STATE, REPORTING, I_ACTION, PERCEPTION and ASPECTUAL.

In the case of multi-token events, we considered only the head of events in building the classification models. Once the events have been extracted and classified, we post-process the text to detect the full extent of multi-token events. The post-processing is done by using the list of multi-token expressions in Italian provided by the task organizers.

The classification models are built using Yamcha. The following features are taken into consideration both for event extent and class identification:

- Token's lemma, stem, PoS tags, flat constituent (noun phrase or verbal phrase), and the entity's type if the token is part of a named entity;
- Whether the token is part of a time expression (labels from the Timex Extraction system);
- Token's simplified PoS (e.g. `n` for nouns, `v` for verbs, etc.), tense for verbs;
- Token's suffix if it is one of the following: -zione, -mento, -tura and -aggio;
- The frequency of the token's appearance in an event extent within the training corpus. We have defined three values to represent the frequency: *never* (the token never appears in an event extent), *sometimes* (it appears more often outside of an event extent than inside), *often* (it appears more often in an event extent than outside);
- Token's WordNet domain;
- Token's derivative if applicable (e.g. *chiudere [close]* for *chiusura [closure]*);
- The preceding 3 labels tagged by the classifier.

The features related to token's suffix, derived word, WordNet domain and frequency are used mainly to improve the recognition of nominal events. The eventive meaning of a noun is indeed difficult to detect with only simple features.

We have submitted three runs that differ from the number of classifiers and the multi-class classification strategy used.

Run 1 / Run2 In both runs two classifiers are used: *(i)* one to identify event extents and *(ii)* one to classify the identified events. For Run 1, the method used for multi-class classification is the one-vs-one strategy, while the one-vs-rest strategy is used for Run 2. All the features described above are used. In addition, some features of the two preceding and the two following tokens are included (e.g. token's PoS, lemma). For event class classification, we have added in the feature set the label predicted by the first classifier (EVENT or O).

Run 3 One single classifier is trained to both detect and classify events. Each token is classified into one of the seven event classes or O for other (i.e. the token is not part of an event extent). The one-vs-rest multi-class classification method is used.

6 Temporal Relation Extraction System

6.1 Temporal Link Identification

In the EVENTI challenge, the task of temporal link identification is restricted to event/event and event/timex pairs within the same sentence. We consider all combinations of event/event and event/timex pairs within the same sentence (in a forward manner) as candidate temporal links. For example, if we have a sentence with an entity order such as "...ev_1...tmx_1...ev_2...", the candidate pairs are (ev_1, tmx_1), (ev_2, tmx_1) and (ev_1, ev_2).

Next, in order to filter the candidate links, we classify a given event/event or event/timex pair into two classes: REL (i.e. the pair is considered as having a temporal link) or O (for other).

A classification model is trained for each type of entity pair (event/event and event/timex), as suggested in previous works (Mani et al., 2006). Again, YamCha is used to build the classifiers. However, this time, a feature vector is built for each pair of entities (e_1, e_2) and not for each token as in the previous classification tasks. The same set of features used for the temporal relation classification task, which are explained in the following section, is applied.

6.2 Temporal Relation Type Classification

Given an ordered pair of entities (e_1, e_2) that could be either event/event or event/timex pair, the classifier has to assign a certain label, namely one of the 13 TimeML temporal relation types: BEFORE, AFTER, IBEFORE, IAFTER, INCLUDES, IS_INCLUDED, MEASURE, SIMULTANEOUS, BEGINS, BEGUN_BY, ENDS, ENDED_BY and IDENTITY.

The classification models are built in the same way as in identifying temporal links. The overall approach is largely inspired by an existing framework for the classification of temporal relations in English documents (Mirza and Tonelli, 2014). The implemented features are as follows:

String and grammatical features. Tokens, lemmas, PoS tags and flat constituent (noun phrase or verbal phrase) of e_1 and e_2, along with a binary feature indicating whether e_1 and e_2 have the same PoS tags (only for event/event pairs).

Textual context. Pair order (only for event/timex pairs, i.e. event/timex or timex/event), textual order (i.e. the appearance order of e_1 and e_2 in the text) and entity distance (i.e. the number of entities occurring between e_1 and e_2).

Entity attributes. Event attributes (*class*, *tense*, *aspect* and *polarity*) [11], and timex *type* attribute [12] of e_1 and e_2 as specified in TimeML annotation. Four binary features are used to represent whether e_1 and e_2 have the same event attributes or not (only for event/event pairs).

Dependency information. Dependency relation type existing between e_1 and e_2, dependency order (i.e. *governor-dependent* or *dependent-governor*), and binary features indicating whether e_1/e_2 is the *root* of the sentence.

Temporal signals. We take into account the list of temporal signals as explained in Section 3.3. Tokens of temporal signals occurring around e_1 and e_2 and and their positions with respect to e_1 and e_2 (i.e. *between* e_1 and e_2, *before* e_1, or at the beginning of the sentence) are used as features.

In order to provide the classifier with more data to learn from, we bootstrap the training data with inverse relations (e.g. BEFORE/AFTER). By switching the order of the entities in a given pair and labelling the pair with the inverse relation type, we roughly double the size of the training corpus.

There are two variations of system submitted.

Run 1 We only consider the frequent relation types, i.e. BEFORE, AFTER, INCLUDES, IS_INCLUDED, MEASURE, SIMULTANEOUS and IDENTITY, in building the classifier for event/event pairs. Using only the frequent relation types results in better performance than using the full set of relation types, because the dataset becomes more balanced.

Run 2 Similar as Run 1, however, we incorporate the TLINK rules for event/timex pairs which conforms to specific signal patterns as explained in the task guidelines[13]. For example, EVENT + dal + DATE$_{type}$ → relType=BEGUN_BY. The event/timex

[11] The event attributes *tense*, *aspect* and *polarity* have been annotated using rules based on the EVENTI guidelines and using the morphological analyses of each token.

[12] The *value* attribute tends to decrease the classifier performance as shown in Mirza and Tonelli (2014), and therefore, it is excluded from the feature set.

[13] http://sites.google.com/site/eventievalita2014/file-cabinet/specificheEvalita_v2.pdf

pairs matching the patterns are automatically assigned with relation types according to the rules, and do not need to be classified.

7 Results

Table 1 shows the results of our system on the two tasks of the EVENTI challenge, i.e. the main task (MT) and the pilot task (PT), and on the 4 subtasks (Task A, B, C and D). For the pilot task we give only the results obtained with the best system.

7.1 Timex Extraction - Task A

For the main task, in recognizing the extent of timex, the system achieves 0.827 F-score using strict-match scheme. The accuracy in determining the timex type is 0.8, while the accuracy in determining the timex value is 0.665.

For the pilot task, in recognizing the extent of timex, the system achieves comparable scores with the main task. However, in determining the timex type and value, the accuracies drop considerably.

7.2 Event Extraction - Task B

On task B the best results are achieved with Run 1, with a strict F-score of 0.867 for event detection and an F-score of 0.671 for event classification. In this run we trained two classifiers using the one-vs-one multi-class classification strategy. On the pilot task data the results are a little bit lower, with a strict F-score of 0.834 for event detection and an F-score of 0.604 for event classification.

Note that for Run 3 due to a problem while training the model on all the training data, we have re-trained the model on only 80% of the data.

7.3 Determining Temporal Relation Types - Task D

For the main task, note that there is a slight error in the format conversion for Run 2. Hence, we recomputed the scores of *Run 2** independently, which results in a slightly better performance compared with Run 1. The system (*Run 2**) yields 0.738 F-score using TempEval-3 evaluation scheme.

For the pilot task (post-submission evaluation), both Run 1 and Run 2 have exactly the same scores, which are 0.588 F-score using TempEval-3 evaluation scheme. This suggests that in the pilot data there is no event/timex pair matching the EVENT-signal-TIMEX3 pattern rules listed in the task guidelines.

7.4 Temporal Awareness - Task C

For this task, we combine the timex extraction system, the 3 system runs for event extraction (Ev), the system for identifying temporal links, and the 2 system runs for classifying temporal relation types (Tr). We found that for both main task and pilot task, the best performing system is the combination of the best run of task B (Ev Run 1) and the best run of task D (Tr Run 1), with 0.341 F-score and 0.232 F-score respectively (strict-match evaluation).

8 Discussion

We have developed an end-to-end system for temporal processing of Italian text. In the EVENTI challenge, we have tested our system on recent newspaper articles, taken from the same sources as the training data, as well as on newspaper articles published in 1914. Without any specific adaptation to historical text, our system yields comparable results.

For the timex extraction task, in identifying the extent and the type of timex, the system achieves good results. In normalizing the timex value, however, the performance is still considerably lower than the state-of-the-art system for English (TimeNorm). This suggests that the TimeNorm adaptation for Italian can still be improved.

For determining timex types and values (as well as temporal relation types), the system performs better on the main task than on the pilot task. With the assumption that the articles written with a gap of one century differ more at the lexical level than at the syntactic level, our take on this phenomena is that in determining timex types, timex values and temporal relation types, the system relies more on the lexical/semantic features. Hence, the performances of the system decrease when it is applied on historical texts.

In the event extraction task, we observed that the event classification performed better with the one-vs-one multi-class strategy than with the one-vs-rest one. Looking at the number of predicted events with both classifiers, the second classifier did not classify all the events found (1036 events were not classified). For this reason the precision is slightly better but the recall is much lower. We have also observed some problems in the detection of multi-token events.

For the relation classification task, as the dataset is heavily skewed, we have decided to reduce the set of temporal relation types. It would be inter-

Subtask	Task	Run	F1	R	P	Strict F1	Strict R	Strict P	type F1	value F1
Task A	**MT**	R1	**0.886**	0.841	0.936	**0.827**	0.785	0.873	0.800	0.665
	PT	R1	**0.870**	0.794	0.963	**0.746**	0.680	0.825	0.678	0.475
Task B	**MT**	R1	**0.884**	0.868	0.902	**0.867**	0.850	0.884	0.671	
		R2	0.749	0.632	0.917	0.732	0.618	0.897	0.632	
		R3	0.875	0.838	0.915	0.858	0.822	0.898	0.670	
	PT	R1	**0.843**	0.793	0.900	**0.834**	0.784	0.890	0.604	
Task D	**MT**	R1	0.736	0.731	0.740	0.731	0.727	0.735		
		R2	0.419	0.541	0.342	0.309	0.307	0.311		
		*R2**	***0.738***	*0.733*	*0.742*	***0.733***	*0.729*	*0.737*		
	PT	*R1 & R2*	***0.588***	*0.588*	*0.588*	***0.570***	*0.570*	*0.570*		
Task C	**MT**	Ev R1 / Tr R1	**0.264**	0.238	0.296	**0.341**	0.308	0.381		
		Ev R1 / Tr R2	0.253	0.241	0.265	0.325	0.313	0.339		
		Ev R2 / Tr R1	0.209	0.167	0.282	0.267	0.209	0.368		
		Ev R2 / Tr R2	0.203	0.168	0.255	0.258	0.212	0.329		
		Ev R3 / Tr R1	0.247	0.211	0.297	0.327	0.279	0.395		
		Ev R3 / Tr R2	0.247	0.211	0.297	0.327	0.279	0.395		
	PT	Ev R1 / Tr R1	**0.185**	0.139	0.277	**0.232**	0.173	0.349		

Table 1: FBK-HLT-time results (MT: Main Task; PT: Pilot Task; Ev Rn: run n of Task B; Tr Rn: run n of Task D)

esting to see if using patterns or trigger lists as a post-processing step can improve the system on the detection of the under-represented relations. For example, the relation type IAFTER (as a special case of the relation AFTER) can be recognized through the adjective *immediato [immediate]*.

In a close future, our system will be included in the TextPro tools suite, both for Italian and English.

Acknowledgments

The research leading to this paper was partially supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404).

References

Steven Bethard. 2013. A synchronous context free grammar for time normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 821–826, Seattle, Washington, USA.

Tommaso Caselli, Hector Llorens, Borja Navarro-Colorado, and Estela Saquete. 2011. Data-driven approach using semantics for recognizing and classifying timeml events in italian. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 533–538, Hissar, Bulgaria.

Hector Llorens, Leon Derczynski, Robert Gaizauskas, and Estela Saquete. 2012. Timen: An open temporal expression normalisation resource. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 753–760, Stroudsburg, PA, USA.

Paramita Mirza and Sara Tonelli. 2014. Classifying Temporal Relations with Simple Features. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 308–317, Gothenburg, Sweden.

Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The TextPro Tool Suite. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

Jannik Strötgen, Julian Zell, and Michael Gertz. 2013. Heideltime: Tuning english and developing spanish resources for tempeval-3. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, pages 15–19, Atlanta, Georgia, USA.

Jannik Strötgen, Ayser Armiti, Tran Van Canh, Julian Zell, and Michael Gertz. 2014. Time for more languages: Temporal tagging of arabic, italian, spanish, and vietnamese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):1–21.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, pages 1–9, Atlanta, Georgia, USA.

10.12871/clicit201429

Overview of the Evalita 2014 SENTIment POLarity Classification Task

Valerio Basile
University of Groningen
v.basile@rug.nl

Andrea Bolioli
CELI, Turin
abolioli@celi.it

Malvina Nissim
University of Groningen
University of Bologna
m.nissim@rug.nl

Viviana Patti
University of Turin
patti@di.unito.it

Paolo Rosso
Universitat Politècnica de València
prosso@dsic.upv.es

Abstract

English. The SENTIment POLarity Classification Task (SENTIPOLC), a new shared task in the Evalita evaluation campaign, focused on sentiment classification at the message level on Italian tweets. It included three subtasks: *subjectivity classification*, *polarity classification*, and *irony detection*. SENTIPOLC was the most participated Evalita task with a total of 35 submitted runs from 11 different teams. We present the datasets and the evaluation methodology, and discuss results and participating systems.

Italiano. *Descriviamo modalità e risultati della campagna di valutazione di sistemi di sentiment analysis (SENTIment POLarity Classification Task), proposta per la prima volta a "Evalita–2014: Evaluation of NLP and Speech Tools for Italian". In SENTIPOLC è stata valutata la capacità dei sistemi di riconoscere il* sentiment *espresso nei messaggi Twitter in lingua italiana. Sono stati proposti tre sotto-task:* subjectivity classification*,* polarity classification *e un sotto-task pilota di* irony detection*. La campagna ha suscitato molto interesse e ricevuto un totale di* 35 *run inviati da* 11 *gruppi di partecipanti.*

1 Introduction

The huge amount of information streaming from online social networking and micro-blogging platforms such as Twitter, is increasingly attracting the attention of researchers and practitioners. The fact that the over 30 teams participated in the Semeval 2013 shared task on Sentiment Analysis in English tweets (Nakov et al., 2013) is indicative in itself.

Several frameworks for detecting sentiments and opinions in social media have been developed for different application purposes, and Sentiment Analysis (SA) is recognized as a crucial tool in social media monitoring platforms providing business services. Extracting sentiments expressed in tweets has been used for several purposes: to monitor political sentiment (Tumasjan et al., 2011), to extract critical information during times of mass emergency (Verma et al., 2011), to detect moods and happiness in a given geographical area from geotagged tweets (Mitchell et al., 2013), and in several social media monitoring services.

Overall, the linguistic analysis of social media has become a relevant topic of research, naturally relying on resources such as sentiment annotated datasets, sentiment lexica, and the like. However, the availability of resources for languages other than English is usually rather scarce, and this holds for Italian as well (Basile and Nissim, 2013; Bosco et al., 2013). The organisation of the SENTIPOLC shared task, articulated in three sub-tasks, was thus aimed at providing reliably annotated data as well as promoting the development of systems towards a better understanding and processing of how sentiment is conveyed in tweets.

2 Task description

The main goal of SENTIPOLC is sentiment analysis at the message level on Italian tweets. We devised three sub-tasks, with increasing complexity.

Task 1: Subjectivity Classification: *a system must decide whether a given message is subjective or objective.*

This is a standard task on recognising whether a message is subjective or objective. (Bruce and Wiebe, 1999; Pang and Lee, 2008).

Task 2: Polarity Classification: *a system must decide whether a given message is of positive, negative, neutral or mixed sentiment.*

Sentiments expressed in tweets are typically categorized as positive, negative or neutral, but a message can contain parts expressing both positive and negative sentiment (mixed sentiment). Differently from most SA tasks, chiefly the Semeval 2013 task, in our data positive and negative polarities are *not* mutually exclusive. This means that a tweet can be at the same time positive *and* negative, yielding a mixed polarity, or also neither positive nor negative, meaning it is a subjective statement with neutral polarity.[1] Section 3.2 provides further explanation and examples.

Task 3 (Pilot): Irony Detection: *a system must decide whether a given message is ironic or not.*

Twitter communications include a high percentage of ironic messages (Davidov et al., 2010; Hao and Veale, 2010; González-Ibáñez et al., 2011; Reyes et al., 2013; Reyes et al., 2014), and platforms monitoring the sentiment in Twitter messages experienced the phenomenon of wrong polarity classification in ironic messages (Bosco et al., 2013). Indeed, the presence of ironic devices in a text can work as an unexpected "polarity reverser" (one says something "good" to mean something "bad"), thus undermining systems' accuracy. In order to investigate this issue, our dataset includes ironic messages, and we devised a pilot subtask concerning irony detection.

The three tasks are meant to be completely independent. For example, a team could take part in the polarity classification task, which only applies to subjective tweets, without tackling Task 1. For each task, each team could submit two runs:

- **constrained**: using the provided training data only; other resources, such as lexicons are allowed; however, it is not allowed to use additional training data in the form of tweets or sentences with sentiment annotations;
- **unconstrained**: using additional data for training, as more sentiment annotated tweets.

Participants willing to submit an unconstrained run for a given task were required to also submit a constrained run for the same task.

3 Development and Test Data

3.1 Corpora Description

The data that we are using for this shared task is a collection of tweets derived from two existing corpora, namely SENTI-TUT (Bosco et al., 2013) and TWITA (Basile and Nissim, 2013). Both corpora have been revised according to the new annotation guidelines specifically devised for this task (see Section 3.3 for details).

There are two main components of the data: a *generic* and a *political* collection. The latter has been extracted exploiting specific keywords and hashtags marking political topics, while the former is composed of random tweets on any topic. Each tweet is thus also marked with a "topic" tag.

A tweet is represented as a sequence of comma-separated fields, namely the Twitter id, the subjectivity field, the positive polarity field, the negative polarity field, the irony field, and the topic field. Apart from the id, which is a string of numeric characters, the value of all the other fields can be either "0" or "1". For the four classes to annotate, 0 and 1 mean that the feature is absent/present, respectively. For the topic field, 0 means "generic" and 1 means "political".

3.2 Manual annotation

The fields with manually annotated values are: `subj`, `pos`, `neg`, `iro`. While these classes could be in principle independent of each other, the following constraints hold in our annotation scheme:

- An objective tweet will not have any polarity nor irony, thus if `subj` $= 0$, then `pos` $= 0$, `neg` $= 0$, and `iro` $= 0$.
- A subjective tweet can exhibit at the same time positive *and* negative polarity (mixed), thus `pos` $= 1$ and `neg` $= 1$ can co-exist.
- A subjective tweet can exhibit no specific polarity and be just neutral but with a clear subjective flavour, thus `subj` $= 1$ and `pos` $= 0$ and `neg` $= 0$ is a possible combination.
- An ironic tweet is always subjective and it must have one defined polarity, so that `iro` $= 1$ cannot be combined with `pos` and `neg` having the same value.

Table 1 summarises the combinations allowed in our annotation scheme. Information regarding manual annotation and the possible combinations was made available to the participants when the development set was released.

The SENTI-TUT section of the dataset was previously annotated for polarity and irony[2]. The tags

[1] In accordance with (Wiebe et al., 2005).

[2] For the annotation process and inter-annotator agreement for the TW–NEWS and TW–FELICITTA portions of

Table 1: Combinations of values allowed by our annotation scheme

subj	pos	neg	iro	description
0	0	0	0	an objective tweet example: *l'articolo di Roberto Ciccarelli dal manifesto di oggi* `http://fb.me/1BQVy5WAk`
1	0	0	0	a subjective tweet with neutral polarity and no irony example: *Primo passaggio alla #strabrollo ma secondo me non era un iscritto*
1	1	0	0	a subjective tweet with positive polarity and no irony example: *splendida foto di Fabrizio, pluri cliccata nei siti internazionali di Photo Natura* `http://t.co/GWoZqbxAuS`
1	0	1	0	a subjective tweet with negative polarity and no irony example: *Monti, ripensaci: l'inutile Torino-Lione inguaia l'Italia: Tav, appello a Mario Monti da Mercalli, Cicconi, Pont...* `http://t.co/3CazKS7Y`
1	1	1	0	a subjective tweet with positive and negative polarity (mixed polarity) and no irony example: *Dati negativi da Confindustria che spera nel nuovo governo Monti. Castiglione: "Avanti con le riforme"* `http://t.co/kIKnbFY7`
1	1	0	1	a subjective tweet with positive polarity, and an ironic twist example: *Letta: sicuramente non farò parte del governo Monti . e siamo un passo avanti. #finecorsa*
1	0	1	1	a subjective tweet with negative polarity, and an ironic twist example: *Botta di ottimismo a #lInfedele: Governo Monti, o la va o la spacca.*

POS, NEG, MIXED and NONE[3] in Senti–TUT were automatically mapped in the following values for the SENTIPOLC's `subj`, `pos`, `neg`, and `iro` annotation fields: POS $\Rightarrow$ 1100; NEG $\Rightarrow$ 1010; MIXED $\Rightarrow$ 1110; NONE $\Rightarrow$ 0000. However, the original Senti–TUT annotation scheme did only partially match the one proposed for this task, in particular regarding the ironic tweets, which were annotated just as HUM in SENTI–TUT, without polarity. Thus, for each tweet tagged as HUM (ca. 800 tweets), two annotators independently added the polarity dimension. The inter-annotator agreement at this stage was $\kappa = 0.259$. In a second round, a third annotator attempted to solve the disagreements (ca. 33%). Tweets where all three annotators had a different opinion (ca. 10%) were discussed jointly for the final label assignment. Note that all the HUM cases that showed no or mixed polarity were considered simply humorous rather than ironic, and marked as 1000 or 1110, respectively.

The TWITA section of the dataset had to be completely re-annotated, as irony annotation was missing, and the three labels adopted in the original data (positive, negative, and neutral, where neutral stood both for objective tweets and subjective tweets with mixed polarity, see (Basile and Nissim, 2013)), were not directly transferrable to the new scheme. The annotation was performed by four experts in three rounds. Round one saw two annotators independently mark each tweet. Inter-annotator agreement was measured at $\kappa = .482$ for Task 1, $\kappa = 0.678$ for positive labels and $\kappa = 0.638$ for negative labels in Task 2, and at $\kappa = 0.353$ for Task 3. In round two, a third annotator made a decision on the disagreements from round one, and in round three a fourth annotator had to decide on those cases where disagreements were left by the previous two rounds. Tweets where all four annotators had a different opinion amounted to just nine cases, and were discussed jointly for the final label assignment.

Finally, to ensure homogenous annotation over the whole dataset, annotators of one subset checked the annotation of the other. No divergences in the guidelines' interpretation surfaced.

3.3 Distribution and data format

Participants were provided with a development set (SentiDevSet henceforth), consisting of 4,513 tweets encoded as described in 3.2. The dataset is the same for all three subtasks.

Due to Twitter's privacy policy, tweets cannot be distributed directly, so participants were also provided with a web interface based on the use of RESTful Web API technology, through which they could download the tweet's text on the fly for all the ids provided.[4]

However, some tweets for which ids were distributed, might be not available anymore at download time for various reasons: Twitter users can

SENTI–TUT see (Bosco et al., 2013; Bosco et al., 2014).

[3]Four annotators collectively reconsidered the set of tweets tagged by NONE in order to distinguish the few cases of subjective, neutral, not-ironic tweets (1000). The original Senti–TUT scheme did not allow such finer distinction.

[4]`http://www.di.unito.it/~tutreeb/sentipolc-evalita14/tweet.html`.

delete their own posts anytime; their accounts can be temporarily suspended or deactivated. As a consequence, it is possible that the number of the available messages in the development dataset will vary over time. In order to deal with this issue, at submission time participants were asked to equip their runs with the information about the number of tweets actually retrieved from SentiDevSet.

The format of the dataset provided by the Web interface is as follows:

"id","subj","pos","neg","iro","top","text"

where the field `text` is to be filled using the procedure available on the website mentioned above. In cases where the tweet is no longer available, the `text` field is filled by the string: "Tweet Not Available", rather than by the text of the tweet.

The version of the data of the SentiDevSet includes for each tweet the manual annotation for the `subj`, `pos`, `neg` and `iro` fields, according to the format explained above. Instead, the blind version of the data for the test set (SentiTestSet henceforth) only contains values for the `idtwitter` and `top` fields. In other words, the development data contains the first six columns annotated, while the test data contains values only in the first (id) and last (topic) columns. In both cases, the `idtwitter` allows to fetch the Twitter message. The distribution of combinations in both datasets is given in Table 2.

Table 2: Distribution of labels in gold standard

combination	SentiDevSet	SentiTestSet
0 0 0 0	1276 (28%)	501 (26%)
1 0 0 0	270 (6%)	111 (6%)
1 0 1 0	1182 (26%)	546 (28%)
1 0 1 1	493 (11%)	209 (11%)
1 1 0 0	895 (20%)	425 (22%)
1 1 0 1	71 (2%)	27 (1%)
1 1 1 0	326 (7%)	116 (6%)
total	4513 (100%)	1935 (100%)

4 Evaluation

4.1 Task1: subjectivity classification

Systems are evaluated on the assignment of a 0 or 1 value to the subjectivity field. A response is considered plainly correct or wrong when compared to the gold standard annotation. We compute precision, recall and F-score for each class (`subj,obj`):

$$precision_{class} = \frac{\#correct\_class}{\#assigned\_class}$$

$$recall_{class} = \frac{\#correct\_class}{\#total\_class}$$

$$F_{class} = 2\frac{precision_{class} recall_{class}}{precision_{class}+recall_{class}}$$

The overall F-score will be the average of the F-scores for subjective and objective classes: $(F_{subj} + F_{obj})/2$

4.2 Task2: polarity classification

Our coding system allows for four combinations of `positive` and `negative` values: 10 (positive polarity), 01 (negative polarity), 11 (mixed polarity), 00 (no polarity). Accordingly, we evaluate positive polarity and negative polarity independently by computing precision, recall and F-score for both classes (0 and 1):

$$precision^{pos}_{class} = \frac{\#correct^{pos}\_class}{\#assigned^{pos}\_class}$$

$$precision^{neg}_{class} = \frac{\#correct^{neg}\_class}{\#assigned^{neg}\_class}$$

$$recall^{pos}_{class} = \frac{\#correct^{pos}\_class}{\#total^{pos}\_class}$$

$$recall^{neg}_{class} = \frac{\#correct^{neg}\_class}{\#total^{neg}\_class}$$

$$F^{pos}_{class} = 2\frac{precision^{pos}_{class} recall^{pos}_{class}}{precision^{pos}_{class}+recall^{pos}_{class}}$$

$$F^{neg}_{class} = 2\frac{precision^{neg}_{class} recall^{neg}_{class}}{precision^{neg}_{class}+recall^{neg}_{class}}$$

The F-score for the two polarity classes is the average of the F-scores of the respective pairs:

$F^{pos} = (F^{pos}_0 + F^{pos}_1)/2$

$F^{neg} = (F^{neg}_0 + F^{neg}_1)/2$

Finally, the overall F-score for Task 2 is given by the average of the F-scores of the two polarities:

$$F = (F^{pos} + F^{neg})/2$$

4.3 Task3: irony detection

Systems are evaluated on their assignment of a 0 or 1 value to the irony field. A response is considered fully correct or wrong when compared to the gold standard annotation. We measure precision, recall and F-score for each class (`ironic,non-ironic`):

$$precision_{class} = \frac{\#correct\_class}{\#assigned\_class}$$

$$recall_{class} = \frac{\#correct\_class}{\#total\_class}$$

$$F_{class} = 2\frac{precision_{class} recall_{class}}{precision_{class}+recall_{class}}$$

The overall F-score will be the average of the F-scores for ironic and non-ironic classes: $(F_{ironic} + F_{non-ironic})/2$

5 Participants and Results

A total of 11 teams from four different countries participated in at least one of the three tasks of SENTIPOLC. Table 3 provides an overview of the teams, their affiliation, and the number of tasks they took part in, with how many runs in total.

Almost all teams participated to both subjectivity and polarity classification subtasks. Most of the submissions were constrained: 9 out of 12 for subjectivity classification; 11 out of 14 for polarity classification; 7 out of 9 for irony detection. In particular, three teams (uniba2930,UNITOR,IRADABE) participated with both a constrained and an unconstrained run on the subtasks of interest. Unconstrained systems did not show to improve performance, but actually decreased it, with the exception of UNITOR's systems, whose unconstrained runs performed better than the constrained ones.

Because of the downloading procedure which we had to implement to comply to Twitter's policies (described in Sec. 3.3), not all teams necessarily tested their systems on the same set of tweets. Differences turned out to be minimal, but to ensure evaluation was performed over an identical dataset for all, we evaluated all participating systems on the union of their classified tweets, which amounted to 1734 (1930-196) [5].

We produced a single-ranking table for each subtask, where unconstrained runs are properly marked. Notice that we only use the final F-score for global scoring and ranking. However, systems that are ranked midway might have excelled in precision for a given class or scored very bad in recall for another. Detailed scores for all classes and all tasks are available in the Appendix.

For each task, we ran a majority class baseline to set a lower-bound for performance. In the tables it is always reported as baseline.

5.1 Task1: subjectivity classification

Table 4 shows results for the subjectivity classification task, which attracted 12 total submissions from 9 teams. The highest F-score was achieved by uniba2930 at 0.7140 (constrained run). All participating systems show an improvement over the baseline.

[5] It turned out that five of the 1935 tweets in SentiTestSet were duplicates.

Table 4: Task 1: F-scores for constrained (F(C)) and unconstrained runs (F(U)).

rank	team	F(C)	F(U)
1	uniba2930	0.7140	0.6892
2	UNITOR	0.6871	0.6897
3	IRADABE	0.6706	0.6464
4	UPFtaln	0.6497	–
5	ficlit+cs@unibo	0.5972	–
6	mind	0.5901	–
7	SVMSLU	0.5825	–
8	fbkshelldkm	0.5593	–
9	itagetaruns	0.5224	–
10	baseline	0.4005	–

5.2 Task2: polarity classification

Table 5 shows results for the polarity classification task, which with 14 submissions from 11 teams was the most popular subtask. Again, the highest F-score was achieved by uniba2930 at 0.6771 (constrained). Also in this case, all participating systems show an improvement over the baseline.[6]

Table 5: Task 2: F-scores for constrained (F(C)) and unconstrained runs (F(U)).

rank	team	F(C)	F(U)
1	uniba2930	0.6771	0.6638
2	IRADABE	0.6347	0.6108
3	CoLingLab	0.6312	–
4	UNITOR	0.6299	0.6546
5	UPFtaln	0.6049	–
6	SVMSLU	0.6026	–
7	ficlit+cs@unibo	0.5980	–
8	fbkshelldkm	0.5626	–
9	mind	0.5342	–
10	itagetaruns	0.5181	–
11	Itanlp-wafi*	0.5086	–
12	baseline	0.3718	–
	*amended run	0.6637	–

5.3 Task3: irony detection

Table 6 shows results for the irony detection task, which attracted 9 submissions from 7 teams. The highest F-score was achieved by UNITOR at 0.5959 (unconstrained run) and 0.5759 (constrained run). While all participating systems show an improvement over the baseline, this time some systems score very close to it, highlighting the complexity of the task.

[6] After the task deadline, the Itanlp-wafi team reported about an error of the conversion script from their internal format to the official one. They submitted, then, the correct run. Official ranking was not revised, but the evaluation of the correct run is shown in the table (marked by star symbol).

Table 3: Teams participating to SENTIPOLC

team	institution	country	tasks	runs
CoLingLab	CoLing Lab – University of Pisa	IT	T2	1
IRADABE	U Politecnica de Valencia / U Paris 13	ES/FR	T1,T2,T3	6
SVMSLU	Minsk State Linguistic University	BY	T1,T2,T3	3
UNITOR	University of Roma Tor Vergata	IT	T1,T2,T3	6
UPFtaln	TALN – Universitat Pompeu Fabra	ES	T1,T2,T3	3
fbkshelldkm	Fondazione Bruno Kessler (FBK-IRST)	IT	T1,T2,T3	3
ficlit+cs@unibo	FICLIT-University of Bologna	IT	T1,T2	2
italianlp-wafi	ItaliaNLP Lab – ILC (CNR)	IT	T2	1
itgetaruns	Ca' Foscari University – Venice	IT	T1,T2,T3	3
mind	University of Milano-Bicocca	IT	T1,T2,T3	3
uniba2930	CS – University of Bari	IT	T1,T2	4

Table 6: Task 3: F-scores for constrained (F(C)) and unconstrained runs (F(U)).

rank	team	F(C)	F(U)
1	UNITOR	0.5759	0.5959
2	IRADABE	0.5415	0.5513
3	SVMSLU	0.5394	–
4	itagetaruns	0.4929	–
5	mind	0.4771	–
6	fbkshelldkm	0.4707	–
7	UPFtaln	0.4687	–
8	baseline	0.4441	–

6 Discussion and Conclusions

We compare the participating systems according to the following main dimensions: exploitation of further Twitter annotated data for training, classification framework (approaches, algorithms, features), exploitation of available resources (e.g. sentiment lexicons, NLP tools, etc.), issues about the interdependency of tasks in case of systems participating in several subtasks.

Most participants restricted themselves to the provided data and submitted constrained systems. Only three teams submitted uconstrained runs, and apart from UNITOR, results are worse than those obtained by the constrained runs. We believe this situation is triggered by the current lack of sentiment-annotated, available large datasets for Italian. Additionally, what might be available is not necessary annotated according to the same principles adopted in SENTIPOLC. Interestingly, uniba2930 attempted acquiring more training data via co-training. They trained two SVM models on SentiDevSet, each with a separate feature set, and then used them to label a large amount of acquired unlabelled data progressively adding training instances to one another's training set, and re-training. No significant improvement was observed, due to the noise introduced by the automatically labelled training instances.

As noticed also in the context of similar evaluation campaigns for the English language (Nakov et al., 2013; Rosenthal et al., 2014), most systems used supervised learning (uniba2930, mind, IRADABE, UNITOR, UPFtaln, SVMSLU, itanlp-wafi, CoLingLab, fbkshelldkm). The most popular algorithm was SVM, but also Decision Trees, Naive Bayes, K-Nearest Neighbors were used. As mentioned, one team experimented with a co-training approach, too.

A variety of features were used, including word-based, syntactic and semantic (mostly lexicon-based) features. The best team in Task1 and Task2, uniba2930, specifically mentions that in leave-one-out experiments, (distributional) semantic features appear to contribute the most. uniba2930 is also the only team that explicitly reports using the topic information as a feature, for their constrained runs. The best team in Task3, UNITOR, employs two sets of features explicitly tailored for the detection of irony, based on emoticons/punctuation and a vector space model to identify words that are out of context. Typical Twitter features were also generally used, such as emoticons, links, usernames, hashtags.

Two participants did not adopt a learning approach. ficlit+cs@unibo developed a system based on a sentiment lexicon that uses the polarity of each word in the tweet and the idea of "polarity intensifiers". A syntactic parser was also used to account for polarity inversion cases such as negations. itgetaruns was the only system solely based on deep linguistic analysis exploiting rhetorical relations and pragmatic insights.

Almost all participants relied on various sentiment lexicons. At least six teams (uniba2930, UPFtaln, fbkshelldkm, ficlit+cs@unibo, UNITOR, IRADABE) used information from Senti-

WordNet (Esuli et al., 2010), either using the already existing Sentix (Basile and Nissim, 2013) or otherwise. Several other lexica and dictionaries were used, either natively in Italian or translated from English (e.g. AFINN, Hu-Liu lexicon, Whissel's Dictionary). Native tools for Italian were used for pre-processing, such as tokenisers, POS-taggers, and parsers.

The majority of systems participating in more than one subtask adopted classification strategies including some form of interdependency among the tasks, with different directions of dependency.

Overall, through a first comparative analysis of the systems' behaviour which we can only briefly summarise here due to space constraints, we can make some observations related to aspects specific to the SENTIPOLC tasks. First, ironic expressions do appear to play the role of polarity reversers, undermining the accuracy of sentiment classifiers. Second, recognising mixed sentiment (tweets tagged as 1110) was hard for our participants, even harder than recognising neutral subjectivity (tweets tagged as 1000). Further and deeper investigations will be matter of future work.

To conclude, the fact that SENTIPOLC was the most popular Evalita 2014 task is indicative of the great interest of the NLP community on sentiment analysis in social media, also in Italy.

Acknowledgments

We would like to thank Manuela Sanguinetti, Cristina Bosco, and Marco Del Tredici for their help in annotating the dataset, and Sergio Rabellino (ICT staff, Dipartimento di Informatica, Turin) for his precious technical support. The last author gratefully acknowledges the support of EC WIQ-EI IRSES (Grant No. 269180) and MICINN DIANA-Applications (TIN2012-38603-C02-01).

References

V. Basile and M. Nissim. 2013. Sentiment analysis on Italian tweets. In *Proc. of WASSA 2013*, pages 100–107, NAACL 2013, Atlanta, Georgia.

C. Bosco, V. Patti, and A. Bolioli. 2013. Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT. *IEEE Intelligent Systems, Special Issue on Knowledge-based Approaches to Content-level Sentiment Analysis*, 28(2):55–63.

C. Bosco, L. Allisio, V. Mussa, V. Patti, G. Ruffo, M. Sanguinetti, and E. Sulis. 2014. Detecting happiness in Italian tweets: Towards an evaluation dataset for sentiment analysis in Felicittà. In B. Schuller et al., editors, *Proc. of ESSSLOD 2014*, pages 56–63, LREC 2014, Reykjavik, Iceland.

R. F. Bruce and J. M. Wiebe. 1999. Recognizing Subjectivity: A Case Study in Manual Tagging. *Nat. Lang. Eng.*, 5(2):187–205, June.

D. Davidov, O. Tsur, and A. Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proc. of CoNLL '10*, pages 107–116, Stroudsburg, PA, USA.

A. Esuli, S. Baccianella, and F. Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of LREC'10*. ELRA, May.

R. González-Ibáñez, S. Muresan, and N. Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *Proc. ACL-HLT'11 - Short Papers - Volume 2*, pages 581–586, Stroudsburg, PA, USA.

Y. Hao and T. Veale. 2010. An ironic fist in a velvet glove: Creative mis-representation in the construction of ironic similes. *Minds Mach.*, 20(4):635–650.

L. Mitchell, M. R. Frank, K. D. Harris, P. S. Dodds, and C. M. Danforth. 2013. The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place. *PLoS ONE*, 8(5), 05.

P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in Twitter. In *Proc. of SemEval 2013*, pages 312–320.

B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

A. Reyes, P. Rosso, and T. Veale. 2013. A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation*, 47(1):239–268.

A. Reyes and P. Rosso. 2014. On the Difficulty of Automatically Detecting Irony: Beyond a Simple Case of Negation. *Knowledge and Information Systems*, 40(3):595–614.

S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. 2014. Semeval-2014 Task 9: Sentiment analysis in Twitter. In *Proc. of SemEval 2014*, pages 73–80, Dublin, Ireland.

A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. 2011. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proc. of ICWSM-11*, pages 178–185, Barcelona, Spain.

S. Verma, S. Vieweg, W. Corvey, L. Palen, J. H. Martin, M. Palmer, A. Schram, and K. M. Anderson. 2011. Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. In *Proc. of the 5th International AAAI Conference on Weblogs and Social Media*, 385–392.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Appendix: Detailed results per class for all tasks

Results of task 1

run	rank	Combined F-score	Prec. (0)	Rec. (0)	F-score (0)	Prec. (1)	Rec. (1)	F-score (1)	team
Constrained	1	0.7140	0.6976	0.5271	0.6005	0.8498	0.8064	0.8275	uniba2930
	2	0.6871	0.5768	0.5872	0.5819	0.8582	0.7358	0.7923	UNITOR
	3	0.6706	0.6247	0.4669	0.5344	0.8284	0.7862	0.8067	IRADABE
	4	0.6497	0.6565	0.3868	0.4868	0.8099	0.8155	0.8127	UPFtaln
	5	0.5972	0.4512	0.4449	0.4480	0.8029	0.6974	0.7464	ficlit+cs@unibo
	6	0.5901	0.4115	0.6473	0.5031	0.8484	0.5632	0.6770	mind
	7	0.5825	0.4363	0.4048	0.4200	0.7917	0.7037	0.7451	SVMSLU
	8	0.5593	0.3791	0.5311	0.4424	0.8050	0.5828	0.6761	fbkshelldkm
	9	0.5224	0.3479	0.3026	0.3237	0.7571	0.6883	0.7211	itagetaruns
	10	0.4005	0.0000	0.0000	0.0000	0.7308	0.8861	0.8010	baseline
Unconstrained	1	0.6897	0.6062	0.5491	0.5762	0.8496	0.7617	0.8032	UNITOR
	2	0.6892	0.6937	0.4629	0.5553	0.8317	0.8148	0.8232	uniba2930
	3	0.6464	0.4729	0.7335	0.5750	0.8955	0.5989	0.7178	IRADABE

Results of task 2

			Positive polarity							Negative polarity							
run	rank	Combined F-score	Prec. (0)	Rec. (0)	F-score (0)	Prec. (1)	Rec. (1)	F-score (1)	F-score	Prec. (0)	Rec. (0)	F-score (0)	Prec. (1)	Rec. (1)	F-score (1)	F-score	team
Constrained	1	0.6771	0.8102	0.8364	0.8231	0.7195	0.4162	0.5274	0.6752	0.7474	0.6890	0.7170	0.6882	0.5995	0.6408	0.6789	uniba2930
	2	0.6347	0.7782	0.8547	0.8147	0.7265	0.2998	0.4245	0.6196	0.7067	0.7107	0.7086	0.6822	0.5213	0.5910	0.6498	IRADABE
	3	0.6312	0.7976	0.7806	0.7890	0.5810	0.4109	0.4814	0.6352	0.6923	0.6701	0.6810	0.6384	0.5201	0.5732	0.6271	CoLingLab
	4	0.6299	0.7949	0.7704	0.7824	0.5604	0.4092	0.4730	0.6277	0.7225	0.6013	0.6564	0.6138	0.6018	0.6078	0.6321	UNITOR
	5	0.6049	0.7782	0.8004	0.7892	0.5766	0.3386	0.4267	0.6079	0.6804	0.6079	0.6421	0.5909	0.5351	0.5616	0.6019	UPFtaln
	6	0.6026	0.7943	0.7337	0.7628	0.5126	0.4303	0.4679	0.6153	0.6627	0.6239	0.6427	0.5856	0.4960	0.5371	0.5899	SVMSLU
	7	0.5980	0.8223	0.5943	0.6899	0.4373	0.5785	0.4981	0.5940	0.6546	0.7663	0.7060	0.6876	0.3901	0.4978	0.6019	ficlit+cs@unibo
	8	0.5626	0.7511	0.8525	0.7986	0.6277	0.2081	0.3126	0.5556	0.6573	0.5495	0.5986	0.5472	0.5339	0.5405	0.5695	fbkshelldkm
	9	0.5342	0.7403	0.7528	0.7465	0.4097	0.2522	0.3122	0.5293	0.6141	0.6089	0.6115	0.5300	0.4166	0.4665	0.5390	mind
	10	0.5181	0.7297	0.8158	0.7703	0.4313	0.1605	0.2339	0.5021	0.6097	0.7700	0.6805	0.6203	0.2819	0.3877	0.5341	itagetaruns
	11	0.5086	0.8106	0.4365	0.5675	0.3636	0.6420	0.4643	0.5159	0.7722	0.2620	0.3913	0.4989	0.7894	0.6114	0.5013	Itanlp-wafi*
	12	0.3718	0.7101	0.9039	0.7954	0.0000	0.0000	0.0000	0.3977	0.5573	0.9114	0.6917	0.0000	0.0000	0.0000	0.3459	baseline
		0.6637	0.8144	0.8048	0.8096	0.6521	0.4462	0.5298	0.6697	0.7287	0.6682	0.6971	0.6614	0.5800	0.6180	0.6576	*amended run
Unconstrained	1	0.6638	0.8189	0.7696	0.7935	0.5969	0.4780	0.5309	0.6622	0.7400	0.6654	0.7007	0.6658	0.5984	0.6303	0.6655	uniba2930
	2	0.6546	0.8212	0.7748	0.7973	0.6080	0.4815	0.5374	0.6673	0.7378	0.5994	0.6615	0.6208	0.6237	0.6223	0.6419	UNITOR
	3	0.6108	0.8204	0.6266	0.7105	0.4565	0.5556	0.5012	0.6058	0.6822	0.6635	0.6727	0.6266	0.5040	0.5587	0.6157	IRADABE

Results of task 3

run	rank	Combined F-score	Prec. (0)	Rec. (0)	F-score (0)	Prec. (1)	Rec. (1)	F-score (1)	team
Constrained	1	0.5759	0.9312	0.6956	0.7963	0.2675	0.5294	0.3554	UNITOR
	2	0.5415	0.8967	0.7849	0.8371	0.2400	0.2521	0.2459	IRADABE
	3	0.5394	0.8990	0.7630	0.8254	0.2274	0.2857	0.2533	SVMSLU
	4	0.4929	0.8829	0.7754	0.8257	0.1566	0.1639	0.1602	itagetaruns
	5	0.4771	0.8933	0.6235	0.7344	0.1570	0.3655	0.2197	mind
	6	0.4707	0.8766	0.7931	0.8328	0.1176	0.1008	0.1086	fbkshelldkm
	7	0.4687	0.8795	0.8889	0.8842	0.2800	0.0294	0.0532	UPFtaln
	8	0.4441	0.8772	0.8995	0.8882	0.0000	0.0000	0.0000	baseline
Unconstrained	1	0.5959	0.9208	0.7630	0.8345	0.3063	0.4286	0.3573	UNITOR
	2	0.5513	0.9139	0.7086	0.7983	0.2387	0.4202	0.3044	IRADABE

10.12871/clicit2014210

UNIBA at EVALITA 2014-SENTIPOLC Task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features

Pierpaolo Basile and **Nicole Novielli**
Department of Computer Science, University of Bari Aldo Moro
Via, E. Orabona, 4 - 70125 Bari (Italy)
{pierpaolo.basile,nicole.novielli}@uniba.it

Abstract

English. This paper describes the UNIBA team participation in the SENTIPOLC task at EVALITA 2014. We propose a supervised approach relying on keyword, lexicon and micro-blogging features as well as representation of tweets in a word space. Our system ranked 1st in both the subjectivity and polarity detection subtasks. As a further contribution, we participated in the unconstrained run, investigating the use of co-training to automatically enrich the labelled training set.

Italiano. *Questo articolo riporta i risultati della partecipazione del team UNIBA al task SENTIPOLC di EVALITA 2014. L'approccio supervisionato che abbiamo proposto affianca alle keyword la rappresentazione semantica dei tweet in uno spazio geometrico, l'utilizzo di feature tipiche dei micro-blog e di dizionari per la definizione della polarità a priori del lessico dei tweet. Abbiamo sperimentato, inoltre, l'uso del co-training per l'arricchimento del dataset tramite annotazione automatica di nuovi tweet.*

1 Introduction

Sentiment analysis is the study of the subjectivity and polarity (positive vs. negative) of a text (Pang and Lee, 2008). With the worldwide diffusion of social media, a huge amount of textual data has been made available and sentiment analysis on micro-blogging is now regarded as a powerful tool for modelling socio-economic phenomena (O'Connor et al., 2010). Dealing with such informal text poses new challenges due to the presence of slang, misspelled words and micro-blogging features such as hashtags or links.

This paper describes our participation at EVALITA 2014 SENTIment POLarity Classification (SENTIPOLC) task (Basile et al., 2014). We discuss methods and results of our experimental studies for the subjectivity and polarity classification subtasks. SENTIPOLC focuses on Italian texts from Twitter. Data provided for training are annotated according to the subjectivity/objectivity of the content carried by the tweet. Moreover, each tweet is categorized as positive, negative, or neutral. Tweet expressing both positive and negative sentiment are also included.

We build a system based on supervised approaches. For training, we exploit three different kinds of feature based on keywords and micro-blogging properties of tweets, on their representation in a distributional semantic model (Vanzo et al., 2014) and on a sentiment lexicon. The purpose of this study is twofold: (i) we propose a method to represent both the tweets and the polarity classes in the word space; (ii) we automatically develop a sentiment lexicon for the Italian starting form SentiWordNet (Esuli and Sebastiani, 2006). Additionally, we propose an approach that exploits co-training to automatically create labelled tweets using the lexicon extracted from a small set of manually annotated data.

The paper is structured as follows: we introduce our system and report the details about features in Section 2. We describe the evaluation and the system setup in Section 3. We conclude by reporting and discussing results in Section 4.

2 System Description

In this section we provide details about the adopted supervised strategy according to the two kinds of run provided by the organizers. In the first one, the *constrained run*, only the provided training data can be used to build the system, but lexicons are allowed. In the second one, the *unconstrained run*, additional training data can be

included. We investigate several kinds of features, which are thoroughly described in Subsection 2.1. To follow the guidelines, we arrange two settings: constrained and unconstrained. In the constrained setting we extract the features from the training data and run the learning algorithm. In the unconstrained condition it is possible to exploit additional training data, (e.g., other corpora with sentiment annotation). Rather than using further manually annotated tweets, we decide to investigate a co-training approach to automatically add new examples to the training set. Figure 1 sketches how co-training is implemented in our system. *Training data* are represented by two different sets of features: *"Feature set 1"* and *"Feature set 2"*. For each feature set we built a separated training model: *"Model 1"* and *"Model 2"*. Unlabeled data, in our case tweets without polarity annotation, are classified using both models. The class selector chooses between predicted classes exploiting classifier confidence: the class with the highest confidence is chosen and the corresponding label is given to the new tweet. The obtained examples can be used as additional training data.

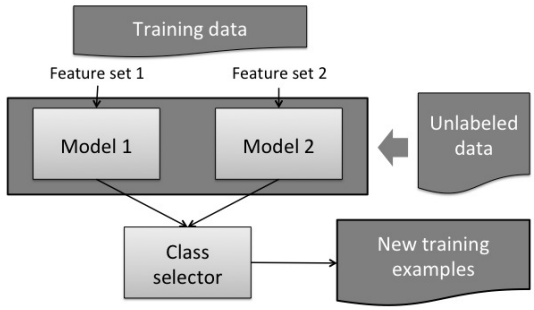


Figure 1: Co-training block diagram.

2.1 Features

We exploit the same features in both settings. In particular, we defined three groups of features based on: (i) keyword and micro-blogging characteristics, (ii) a sentiment lexicon, and (iii) a Distributional Semantic Model (DSM).

Keyword based features exploit tokens occurring in the tweets, only unigrams are considered. During the tokenization we replace the user mentions, URLs and hashtags with three metatokens: "_USER_", "_URL_" and "_TAG_". We create features able to capture several aspects of microblogging, such as the use of upper case and character repetitions[1], positive and negative emoticons, informal expressions of laughters[2], as well as the presence of exclamation and interrogative marks, adversative words[3], disjunctive words[4], conclusive words[5] and explicative words[6].

The second group of features concerns the DSM. Given a set of unlabelled downloaded tweets, we build a geometric space in which each word is represented as a mathematical point. The similarity between words is computed as their closeness in the space. To represent a tweet in the geometric space, we adopt the superposition operator (Smolensky, 1990), that is the vector sum of all the vectors of words occurring in the tweet. We use the tweet vector $\overrightarrow{t}$ as a semantic feature in training our classifiers. In the same fashion, we build also prototype vector for each class as the sum of all the tweet vectors belonging to the given class. We use two prototype vectors to represent, respectively, subjectivity $\overrightarrow{p_s}$ and objectivity $\overrightarrow{p_o}$. Analogously, we build four prototype vectors for positive $\overrightarrow{p_{pos}}$, negative $\overrightarrow{p_{neg}}$, positive and negative $\overrightarrow{p_{pn}}$, and neutral $\overrightarrow{p_n}$ polarity. To capture the subjectivity of a tweet $\overrightarrow{t}$, we add to the DSM features the cosine similarity between $\overrightarrow{t}$ and $\overrightarrow{p_s}$, and the similarity between $\overrightarrow{t}$ and $\overrightarrow{p_o}$. Thus, we compute all the similarity score with respect to the four prototype vectors for polarity.

Finally, the third block contains features extracted from the SentiWordNet (Esuli and Sebastiani, 2006) lexicon. We translate SentiWordNet in Italian through MultiWordNet (Pianta et al., 2002). It is important to underline that SentiWordNet is a synset-based lexicon while our Italian translation is a word based lexicon.

In order to automatically derive our Italian sentiment lexicon from SentiWordNet, we perform three steps. First, we translate the synset offset in SentiWordNet from version 3.0 to 1.6[7] using automatically generated mapping file. Then, we transfer the prior polarity of SentiWordNet to the Italian lemmata. Each synset in SentiWordNet has three polarity scores, negative, positive, and neutral, which are transferred to all the Italian lemmata belonging to the corresponding MultiWord-

[1]These features usually plays the same role of intensifiers in informal writing contexts.

[2]i.e., sequences of "ah".

[3]ma, bensì, però, tuttavia, peraltro, nondimeno, pure, eppure, sennonché, anzi, invece.

[4]o, oppure, ovvero, ossia.

[5]dunque, quindi, perciò, pertanto, onde, sicché.

[6]infatti, cioè, ossia.

[7]Since MultiWordNet is based on WordNet 1.6.

Net synset. By using this approach, a lemma can receive multiple polarity scores if it occurs in more than one synset. In such cases, we assign to the lemma the average polarity score. In the lexicon we add also emoticons as taken from Wikipedia[8]: we assign a positive score equal to 1 to the positive emoticons, and a negative score equal to 1 to the negative ones. Finally, we expand the lexicon using Morph-it! (Zanchetta and Baroni, 2005), a lexicon of inflected forms with their lemma and morphological features. We extend the polarity scores of each lemma to its inflected forms. Our strategy for creating the Italian polarity lexicon is similar to the one adopted in (Basile and Nissim, 2013), which however deal differently with multiple polarity scores for an ambiguous lemma.

The obtained Italian translation of SentiWordNet is used to compute a set of features based on prior polarity of words in the tweets, as reported in Table 3. To deal with mixed polarity cases we defined two sentiment variation features so as to capture the simultaneous expression of positive and negative sentiment in the same tweet.

The complete list and description of microblogging, semantic and lexicon features are reported in Tables 1, 2 and 3, respectively. A boolean feature that indicates if a tweet concerns the politic topic or not is finally added. Since this feature is only present in the training data, we remove it in the unconstrained run.

3 Evaluation

The EVALITA-2014 SENTIPOLC Task is designed for evaluating systems on their ability in: Task 1) decide whether a given tweet is subjective or objective; Task 2) decide the tweet polarity with respect to four classes: positive, negative, neutral and mixed sentiment (both positive and negative).

Organizers provided 4,513 manually annotated tweets as training data. At the time of the evaluation, 495 tweets are not available for the download and are removed from the training. We use the annotated data to extract the features and independently train the classifiers for Tasks 1 and 2. Section 3.1 reports details on our system setup.

As test set, organizers provided a collection of 1,935 manually annotated tweets (1,748 available at the time of the evaluation). Systems are compared against the gold standard in terms of F measure. Results are reported in Section 4.

[8] http://it.wikipedia.org/wiki/Emoticon

3.1 System Setup

The system is completely developed in JAVA, and the Weka[9] library is adopted for the Support Vector Machine[10]. Tweets are tokenized using "Twitter NLP and Part-of-Speech Tagging"[11] API developed by the Carnegie Mellon University. We use only the tokenizer since previous research has shown that part-of-speech features are not crucial for sentiment analysis on tweets (Kouloumpis et al., 2011).

Regarding the DSM, we download 10 million tweets using the Twitter Streaming API. Tweets are downloaded by querying the API using four lexicons extracted from the training data for each class. In particular, tweets in training set are divided in two classes: subjective and objective. For each class we extract a lexicon. Analogously, tweets in training set are divided into positive and negative. We add mixed polarity tweets to both positive and negative classes. Thus, we extract a lexicon for the positive class and a lexicon for the negative one. To extract the lexicons we use a probabilistic approach. We compute the probability for each token as:

$$P(t|c_i) = \frac{\#t + 1}{\#tot_i + |V|} \quad (1)$$

where c_i is the class, $\#t$ are the occurrences of t in c_i, $\#tot_i$ are the total occurrences in c_i, and V is the vocabulary.

For each lexicon, we rank tokens in descending order according to the Kullback-Leibler divergence (KLD). For example, in the case of subjectivity detection, we compute token probabilities for both subjective c_s and objective c_o classes. For each token t in V we calculate the KLD between $P(t|c_s)$ and $P(t|c_o)$ as:

$$KLD = P(t|c_s) * log\frac{P(t|c_s)}{P(t|c_o)} \quad (2)$$

The top terms in the rank are relevant for the c_s class. We perform this computation for each lexicon to extract the most 50 relevant terms for subjective, objective, positive and negative classes. We use these terms as seeds for downloading the same number of tweets for each lexicon.

We exploit these unlabeled new tweets to build a DSM, using the "word2vec"[12] tool based on a re-

[9] http://www.cs.waikato.ac.nz/ml/weka/

[10] We also experimented with Random Forest with comparable performance.

[11] http://www.ark.cs.cmu.edu/TweetNLP/

[12] https://code.google.com/p/word2vec/

Keyword and micro-blogging features	
$n-grams$	only unigrams are considered. User mentions, URLs and hashtag are replaced with metatokens
$count_{USER}$	total occurrences of user mentions
$count_{URL}$	total occurrences of URLs
$count_{TAG}$	total occurrences of hashtags
$upperCase_{ratio}$	the ratio between the number of upper case characters and the total number of characters
emo_{pos}	the number of positive emoticons
emo_{neg}	the number of negative emoticons
$count_{Laugh}$	the count of sequences of 'ah' as slang expression of laughters
$count_{Intensif}$	the ratio between the number of tokens with repeated characters and the total number of tokens
$count_{QMark}$	the total occurrences of question marks
$count_{ExMark}$	the total occurrences of exclamation marks
$count_{advers}$	the total occurrences of adversative words
$count_{disj}$	the total occurrences of disjunctive words
$count_{concl}$	the total occurrences of conclusive words

Table 1: Description of keyword and micro-blogging features.

Semantic features	
$\overrightarrow{t}$	the representation of the tweet vector in the word space
sim_{subj}	the similarity between $\overrightarrow{t}$ and the subjective prototype vector $\overrightarrow{p_s}$
sim_{obj}	the similarity between $\overrightarrow{t}$ and the objective prototype vector $\overrightarrow{p_o}$
sim_{pos}	the similarity between $\overrightarrow{t}$ and the positive prototype vector $\overrightarrow{p_{pos}}$
sim_{neg}	the similarity between $\overrightarrow{t}$ and the negative prototype vector $\overrightarrow{p_{neg}}$
sim_{posneg}	the similarity between $\overrightarrow{t}$ and the mixed polarity prototype vector $\overrightarrow{p_{pn}}$
$sim_{neutral}$	the similarity between $\overrightarrow{t}$ and the neutral prototype vector $\overrightarrow{p_n}$

Table 2: Description of semantic features.

vised implementation of the Recurrent Neural Net Language Model (Mikolov et al., 2013) using a log-linear approach. In particular, we use the Continuous Bag-of-Words Model (CBOW) with 200 vector dimensions. We remove the terms with less than ten occurrences, obtaining a total number of about 200,000 terms overall.

We trained our classifiers using a SVM with the RBF kernel, setting the C parameter to 4. We select these values after a 10-fold validation on training data to select the best combination. The total number of features is 12,117. In the constrained run, the entire set of features is used for both subjectivity and polarity classification tasks. Regarding the unconstrained run, we split the features in two subsets to implement the co-training approach. The first set (Feature set 1 in Figure 1) is composed by keyword and micro-blogging, and lexicon features used to learn Model 1; the second set (Feature set 2) exploits the semantic features to learn Model 2. In the co-training strategy we obtained about 40,000 new examples automatically tagged.

4 Results and Discussion

The overall system performance is assessed in terms of F measure, according to the measure adopted by the task organizers. Table 4 reports the system performance, its rank, and the percentage improvement over the baseline calculated assigning the most frequent class in the gold standard.

The results are very encouraging: the system always obtains the best performance in all settings and in Task 1 of the un-constrained run it differs for only 0.0005 from the first ranked one. We observe that the co-training approach seems

Sentiment lexicon based features	
p_{subj}	the subjectivity polarity, it is the sum of the positive and negative scores
p_{obj}	the objectivity polarity, it is the sum of the neutral scores
o_{subj}	the number of tokens having the positive or negative score higher than zero
o_{obj}	the number of tokens having the neutral score higher than zero
r_{subj}	the ratio between p_{subj}/o_{subj}
r_{obj}	the ration between p_{obj}/o_{obj}
$subjobj_{diff}$	the difference between $r_{subj} - r_{obj}$
sum_{pos}	the sum of positive scores for the tokens in the tweet
sum_{neg}	the sum of negative scores for the tokens in the tweet
o_{pos}	the number of tokens that have the positive score higher than zero
o_{neg}	the number of tokens that have the negative score higher than zero
r_{pos}	the ratio between sum_{pos}/o_{pos}
r_{neg}	the ration between sum_{neg}/o_{neg}
$posneg_{diff}$	the difference between $r_{pos} - r_{neg}$
max_{pos}	the sum of the positive scores, where $positive\ score > negative\ score$
max_{neg}	the sum of the negative scores, where $negative\ score > positive\ score$
max_{subj}	the sum of max_{pos} and max_{neg}
max_{obj}	the sum of the neutral scores, where the neutral score is higher than both the positive and negative ones
$subjobj_{max_{diff}}$	the difference between $max_{subj} - max_{obj}$
$posneg_{max_{diff}}$	the difference between $max_{pos} - max_{neg}$
$sentiment\ variation$	for each token occurring in the tweet a tag is assigned, according to the highest polarity score of the token in the Italian lexicon. Tag values are in the set {OBJ, POS , NEG}. The sentiment variation counts how many switches from POS to NEG, or vice versa, occur in the tweet.
$sentiment\ variation\ pos/neg$	it is similar to the previous feature, but the OBJ tag is assigned only if both positive and negative scores are zero. Otherwise, the POS tag is assigned if the positive score is higher than the negative one, vice versa the NEG tag is assigned.

Table 3: Description of sentiment lexicon features.

Setting	Task	F	Rank	Imp.
baseline	Task 1	0.4005	-	-
	Task 2	0.3718	-	-
constrained	Task 1	0.7140	1	78%
	Task 2	0.6771	1	82%
unconstrained	Task 1	0.6892	2	72%
	Task 2	0.6638	1	79%

Table 4: System results for each task and setting.

to introduce noise and need to be tuned in future replication of our study. A deep analysis of the results shows that the co-training system slightly improves the performance in classifing positive tweets, while the performance in other classes decreases. Details about each class are reported in Table 5, improvements in the un-constrained task are underlined by the ↑ symbol. The evaluation criteria for the polarity task involve consideration of mixed cases as both negative and positive.

After an error analysis, we discover a bias in our classifier due to the domain-specific lexicon about political topics. This is the main cause of error in the classification of the objective tweets, which are labeled as subjective in 58% of misclassified cases due to the presence of lexicon related to topics for which people generally express a negative opinion[13]. For the same reason, the 37% and the 44% of misclassified neutral and positive cases, respectively, are classified as negative. Furthermore, we observe that the recall of our classifier could be improved for both positive and negative classes by enriching our lexicon with jargon and idiomatic expressions. Finally, in the 43% of misclassified negative cases common sense reasoning would be required to detect the negative opinion expressed

[13] e.g., Monti, governo, Grillo.

Setting	Class	False (F)			True (T)			Comb.
		P_F	R_F	F_F	P_T	R_T	F_T	F
Constrained	sub	0.6976	0.5271	0.6005	0.8498	0.8064	0.8275	0.7140
	pos	0.8102	0.8364	0.8231	0.7195	0.4162	0.5274	0.6752
	neg	0.7474	0.6869	0.7170	0.6882	0.5995	0.6408	0.6789
Un-constrained	sub	0.6937	0.4629	0.5553	0.8317	0.8148	0.8232	0.6892
	pos	0.8189	0.7696	0.7935	0.5969	0.4780	0.5309 ↑	0.6622
	neg	0.7400	0.6654	0.7007	0.6658	0.5984	0.6303	0.6655

Table 5: System results for each class.

by the author[14], including ironic tweets.

As a further investigation of the predictive power of the features in our model, we perform an ablation test for both tasks. We removed each group of features to assess the decrease of F measure on test data with respect to the setting including all features. Results are reported in Figures 2 and demonstrate the importance of all feature groups. Particularly, semantic features plays a key role, as we observe how removing them causes the highest decrease in performance in both tasks.

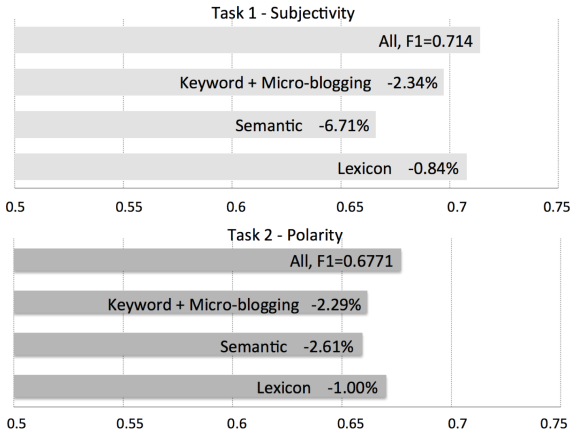


Figure 2: Decrease of F by removing each feature group, compared to the complete feature setting.

Future replication of this study will involve further data, to validate and generalize our findings.

Acknowledgements

This work is partially funded by the ATS Romantic Living Lab under the Apulian ICT Living Labs program and the project PON 01 00850 ASK-Health (Advanced System for the interpretation and sharing of knowledge in health care).

[14]"Governo Monti: ipotesi #Passera allo Sviluppo. Candidatura spontanea della Minetti."

References

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proc. of WASSA 2013*, pages 100–107.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proc. of EVALITA 2014*, Pisa, Italy.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proc. of LREC*, pages 417–422.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D. Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In *Proc. of ICWSM 2011*, pages 538–541.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. of ICLR Work.*

Brendan O'Connor, Ramnath Balasubramanyan, Bryan Routledge, and Noah Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Intl AAAI Conf. on Weblogs and Social Media (ICWSM)*, volume 11, pages 122–129.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proc. 1st Intl Conf. on Global WordNet*, pages 293–302.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216, November.

Andrea Vanzo, Danilo Croce, and Roberto Basili. 2014. A context-based model for sentiment analysis in twitter. In *Proc. of COLING 2014*.

Eros Zanchetta and Marco Baroni. 2005. Morph-it!: a free corpus-based morphological resource for the italian language. *Proc. of the Corpus Linguistics Conf. 2005*.

10.12871/clicit2014211

ITGETARUNS A Linguistic Rule-Based System for Pragmatic Text Processing

Rodolfo Delmonte
Dipartimento di Studi Linguistici e Culturali Comparati
Ca' Bembo – Dorsoduro 1075
Università Ca' Foscari – 30123 VENEZIA, Italy
delmont@unive.it

Abstract

English. We present results obtained by our system ITGetaruns for all tasks. It is a linguistic rule-based system in its bottom-up version that computes a complete parser of the input text. On top of that it produces semantics at different levels which is then used by the algorithm for sentiment and polarity detection. Our results are not remarkable apart from the ones related to Irony detection, where we ranked fourth over eight participants. The results were characterized by our intention to favour Recall over Precision and this is also testified by Recall values for Polarity which in one case rank highest of all.

Italiano. *Presentiamo i risultati ottenuti dal nostro sistema ITGetaruns per tutti i task. Si tratta di un sistema basato su regole linguistiche nella sua versione bottom-up, che produce un parse complete del testo in ingresso. Al di sopra di questo produce semantica a diversi livelli, che viene poi usata dall'algoritmo per l'analisi della polarità e della soggettività. I nostri risultati non sono notevoli a parte quelli relativi alla individuazione dell'Ironia, nella quale ci siamo classificati quarti su sette partecipanti. I risultati sono caratterizzati dalla nostra intenzione di favorire il Recall sulla Precision and questo è anche documentato dai valori della Recall per la polarità che in un caso sono i più alti in assoluto.*

1 Description of the System

The system we called ITGetaruns shares its backbone with the companion English system which has been used – and documented – for a number of international challenges on Semantic and Pragmatic computing in English texts. It is organized around a manually checked subcategorized lexicon, a sequence of rules organized according to theoretical linguistics criteria and combines data-driven (bottom-up) and grammar-driven (top-down) techniques.

Technically speaking, it is based on a shallow parser, which in turn is based on a chunker and NER and multiword recognizer. On top of this parser, there is constituent or phrase structure parser, which sketches sentence structure. This is then passed to a deep dependency parser, which combines constituent level information, lexical information, and a Deep Island Parser. The aim of this third parser is that of producing semantically viable Predicate-Argument Structures. Finally, on top of this level of representation, the Pragmatic System is built.

Conceptually speaking, the deep island parser (hence DIP) is very simple to define, but hard to implement. A semantic island is made up by a set of A/As, which are dependent on a verb complex (hence VCX). Arguments and Adjuncts may occur in any order and in any position: before or after the verb complex, or be simply empty or null. Their existence is determined by constituents surrounding the VCX. The VCX itself can be composed of all main and minor constituents occurring with the verb and contributing to characterize its semantics. We are here referring to: proclitcs, negation and other adverbials, modals, restructuring verbs (*lasciare*/let, *fare*/make, etc.), and all auxiliaries. Tensed morphology can then appear on the main lexical verb or on the auxiliary/ modal/ restructuring verb. Gender can appear on the past participle when the verb takes auxiliary ESSERE, or when a complement is duplicated by Clitic Left Dislocation.

The DIP is preceded by a tagger, which is accompanied by a multiword expression labeller. Tagged input is passed to an augmented context-free parser that works on top of a chunker. The chunker collects main constituents on the basis of a Recursive Transition Network of Italian and then passes the output to a cascaded sentence level parser. Constituents are labelled with usual

grammatical relations on the basis of syntactic subcategorization contained in our verb lexicon of Italian counting some 17,000 entries. There are some 270 different syntactic classes, which differentiates also the most common prepositions associated to oblique arguments. Linear position and precedence in the input string is assumed at first as a valid criterion for distinguishing SUBJects from OBJects. Adjustments will be executed by the semantic parser, which will be responsible for the final relabeling of the output.

The DIP receives the output of the surface parser, a list of Referring Expressions and a list of VCX. Referring expressions are all nominal heads accompanied by semantic class information collected in a previous recursive run through the list of the now lemmatized and morphologically analysed input sentence. It also receives the output of the context-free parser. The DIP searches for SUBJects at first and assumes it is positioned before the verb and close to it. In case there is none such chunk available the search is widened if intermediate chunks are detected: they can be Prepositional Phrases, Adverbials or simply Parentheticals. If this search fails, the DIP looks for OBJects close after the verb then and again possibly separated by some intermediate chunk. They will be relabelled as Subjects. Conditions on the A/As boundaries are formulated in these terms: between current VCX and prospective argument there cannot be any other VCX. Additional constraints regard presence of relative or complement clauses, which are detected from the output chunked structure.

The prospective argument is deleted from the list of Referring Expressions and the same happens with the VCX. The same applies for the OBJect, OBJect1 and OBLique. When arguments are completed, the parser searches recursively for ADJuncts, which are PPs, using the same boundary constraint formulation above.

Special provisions are given to copulative constructions, which can often be reversed in Italian: the predicate coming first and then the subject NP. The choice is governed by looking at referring attributes, which include definiteness, quantification, distinction between proper/common noun. It assigns the most referring nominal to the SUBJect and the less referring nominal to the predicate. In this phase, whenever a SUBJect is not found from available referring expressions, it is created as little_pro and morphological features are added from the ones belonging to the verb complex. After updating of the Referring Expressions with the new Grammatical Relations, the parser searches the most adequate Semantic Role to be associated to it. This is again taken from a lexicon of corresponding verb predicates and works according to the type of overall Predicate-Argument Structure (hence PAS).

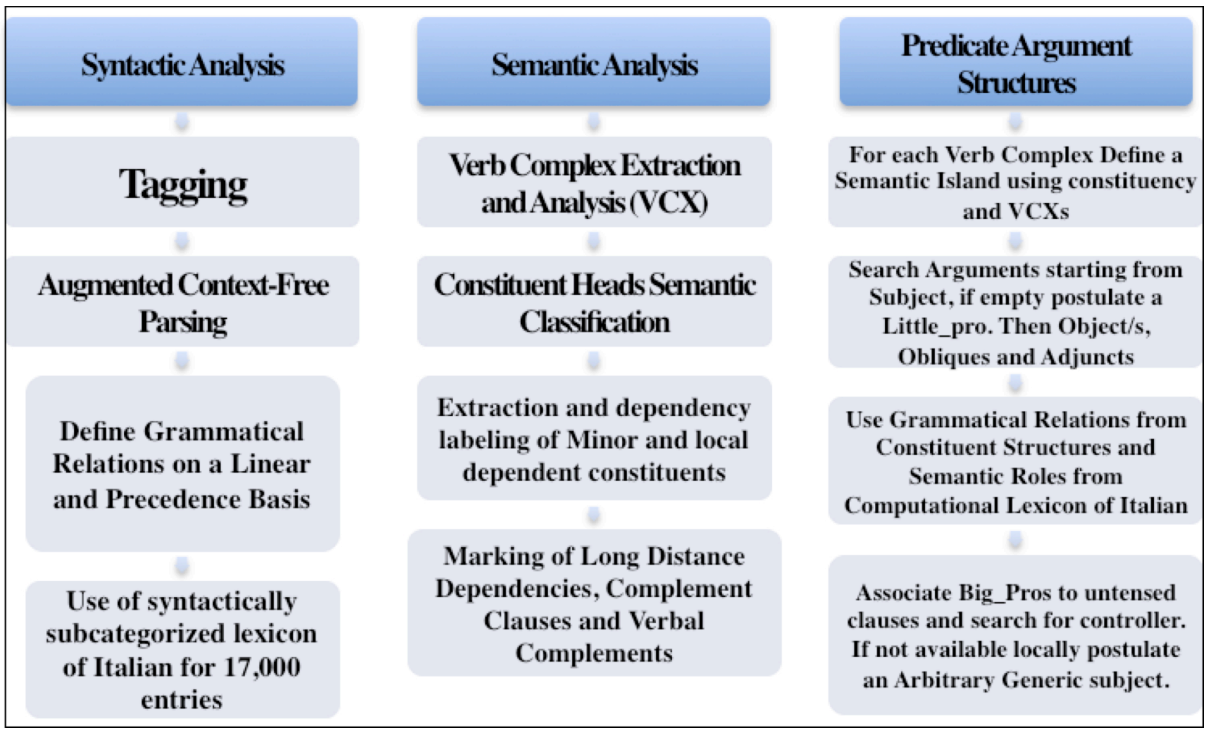


Table 1. Flowchart of modules for Deep Island Parser.

The SUBJect is in fact strictly depending on the semantics associated to the verb, but in case of ambiguity the system delays the assignment of semantic role until a complete PAS is obtained. In this phase, passive diathesis is checked in order to apply a lexical rule from LFG, that assigns OBJect semantic role to the SUBJect of the corresponding passive form of the verb predicate.

The PAS thus obtained, is then enriched by a second part of the algorithm, which adds empty or null elements to untensed clauses. The system starts from little_pros and looks for local possible antecedents. An additional semantic function is activated in this phase of analysis and is the creation of verbal multiwords, constituted by the concatenation of a verb lemma and the head of its object, as for instance "tener conto"/take_into_account, which transforms the main predicate TENER into TENER_CONTO. In this operation, the system has available a list of light verbs of Italian which are the most frequent main component of the compound: then the OBJect complement head is extracted and the concatenation is searched in a specialized dictionary of verbal multiwords of Italian. The OBJect is then erased from the list of arguments and the Argument/Adjunct distinction is updated according to the new governing predicate.

1.1 The Pragmatic Parser

Measuring the polarity of a text is usually done by text categorization methods which rely on freely available resources. However, we assume that in order to properly capture opinion and sentiment (see Delmonte & Pallotta 2011; Kim & Hovy 2004; Pang & Lee 2004; Wiebe et al. 2005), expressed in a text or dialog, - that we also assume to denote the same field of research, and is strictly related to "subjectivity" analysis - any system needs a linguistic text processing approach that aims at producing semantically viable representation at propositional level. In particular, the idea that the task may be solved by the use of Information Retrieval tools like Bag of Words Approaches (BOWs) is insufficient. BOWs approaches are sometimes also camouflaged by a keyword based Ontology matching and Concept search (see Kim and Hovy 2004), based on SentiWordNet (see Esuli & Sebastiani 2006) more on this resource below -, by simply stemming a text and using content words to match its entries and produce some result (Turney and Littman 2003). Any search based on keywords and BOWs is fatally flawed by the impossibility to cope with such fundamental issues as the following ones, which Polanyi & Zaenen (2006) named contextual valence shifters:

- presence of negation at different levels of syntactic constituency;
- presence of lexicalized negation in the verb or in adverbs;
- presence of conditional, counterfactual subordinators;
- double negations with copulative verbs;
- presence of modals and other modality operators.

It is important to remember that both Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA) (Turney & Littman 2003) systematically omit function or stop words from their classification set of words and only consider content words. In order to cope with these linguistic elements we propose to build a propositional level analysis directly from a syntactic constituency or chunk-based representation. We implemented these additions on our system thus trying to come as close as possible to the configuration which has been used for semantic evaluation purposes in challenges like Recognizing Textual Entailment (RTE) and other semantically heavy tasks (see Bos & Delmonte 2008; Delmonte et al. 2010). The output of the system is an xml representation where each sentence of a text or dialog is a list of attribute-value pairs. In order to produce this output, the system makes use of a flat syntactic structure and a vector of semantic attributes associated to the verb compound at propositional level and memorized. An important notion required by the extraction of opinion and sentiment is also the distinction of the semantic content of each proposition into two separate categories: objective vs. subjective.

This is obtained by searching for factivity markers again at propositional level (see Saurì & Pustejovsky 2012). In particular we take into account the following markers: modality operators such as intensifiers and diminishers, modal verbs, modifiers and attributes adjuncts at sentence level, lexical type of the verb (from ItalWordNet classification, and our own), subject's person (if 3rd or not), and so on. As will become clear below, we are using a lexicon-based (see Pennebaker et al.; Taboada et al. 2011) rather than a classifier-based approach, i.e. we make a fully supervised analysis where semantic features are manually associated to lemma and concept of the domain by creating a lexicon out of frequency lists. In this way the semantically labelled lexicon is produced in an empirical manner and fits perfectly the classification needs. Now, the new current version used with Italian has been made possible by the creation of the needed semantic resources, in particular a version of SentiWordNet adapted to

Italian and heavily corrected and modified. This version uses weights for the English WordNet and the mapping of sentiment weights has been done automatically starting from the linguistic content of WordNet glosses. This process has introduced a lot of noise in the final results, with many entries with a totally wrong opinion evaluation. In addition, there was a need to characterize uniquely only those entries that have a "generic" or "commonplace" positive, or negative meaning associated to them in the specific domain. This was deemed the only possible solution to the problem of semantic ambiguity, which could only be solved by introducing a phase of Word Sense Disambiguation, which was not part of the system. However this was not possible for all entries. So, we decided to erase all entries that had multiple concepts associated to the same lemma, and had conflicting sentiment values. We also created and added an ad hoc lexicon for the majority of concepts (some 3000) contained in the texts we analysed, in order to increase the coverage of the lexicon. This was done again with the same approach, i.e. labelling only those concepts which were uniquely intended as one or the other sentiment, restricting reference to the domain of political discourse.

1.2 Semantic Mapping

Sentiment Analysis is based on propositional level semantic processing, which in turn is made of two basic components: PAS and VCX semantics. Semantic mapping is based on a number of intermediate semantic representations, which include, beside diathesis:

- Change in the World; Subjectivity and Point of View; Speech Act; Factuality; Polarity.

At first we compute Mood and Tense from the Verbal Compound (hence VC), which, as said before, may contain auxiliaries, modals, clitics, negation and possibly adverbials in between. From Mood_Tense we derive a label that is the compound tense and this is then used together with Aspectual lexical properties of the main verb to compute Change_in_the_World. Basically this results into a subclassification of events into three subclasses: Static, Gradual, Culminating. From Change_in_the_World we compute (Point_of_) View, which can be either Internal (Extensional/Intensional) or External, where Internal is again produced from a semantic labelling of the subcategorized lexicon along the lines suggested in linguistic studies, where psych(ological) verbs are separated from movement verbs etc. . Internal View then allows a labelling of the VC as Subjective for Subjectivity and otherwise, Objective. Eventually, we look for negation which can be produced by presence of a negative particle or be directly in the verb meaning as lexicalised negation. Negation, View and Semantic Class, together with presence of absence of Adverbial factual markers are then used to produce a Factuality labelling.

One important secondary effect that carries over from this local labelling, is a higher level propositional level ability to determine inferential links intervening between propositions. Whenever we detect possible dependencies between adjacent VCs we check to see whether the preceding verb belongs to the class of implicatives. We are here referring to verbs such as “refuse, reject, hamper, prevent, hinder, etc.” on the one side, and “manage, oblige, cause, provoke, etc.” on the other (for a complete list see Saurì & Pustejovsky 2012). In the first case, the implication is that the action described in the complement clause is not factual, as for instance in “John refused to drive to Boston”, from which we know that “John did not drive to Boston”. In the second case, the opposite will apply, as in “John managed to drive to Boston”.

Two notions have been highlighted in the literature on discourse: foreground and background. The foreground is that part of a discourse which provides the main information; in a narrative, for example, the foreground is the temporal sequence of events; foreground information, then, moves the story forward. The background, on the contrary, provides supportive information, such as elaborations, comments, etc., and does not move the story forward. To compute foreground and background information, three main rhetorical relations are assigned by the algorithm (for a deeper description see Delmonte 2007; 2009) in the form of attribute-value pairs, or features: Discourse Domain, CHANGE IN THE WORLD.

The Discourse Domain of a sentence may be “subjective”, indicating that the event or state takes place in the mind of the participant argument of the predicate and not necessarily in the external world. Then it may be “objective”, which indicates that the action described by the verb affects the whole environment. A sentence may also describe a “change in the world”, in case we pass from the description of one situation to the description of another situation which precedes or follows the former in time but which is not temporally equivalent to it; we have then the following inventory of changes: null (i.e. no change), gradual, culminated, earlier, negated. The third value, the “relevance” of a sentence, corresponds to the distinction between foreground and background which has been discussed above.

We have now to explain the way each utterance receives its set of values: the algorithm relies heavily on grammatical cues, i.e. those linguistic elements encoded in the grammar of a language which allow interpretation without the intervention of pragmatic or non-linguistic elements such as conversational implicatures, presupposition or inferencing. The cues we make use of are chiefly extracted from the verb and are such things as semantic category, polarity, tense, aspect. The procedure is very simple from a theoretical point of view: once the algorithm has recognized a cue, it assigns a value to the sentence. Note that we distinguish between the direct and indirect speech portions of the text, since the perspective is not the same in the two cases.

- DISCOURSE DOMAIN: to assign the point of view of a sentence, the algorithm checks the sem(antic)_cat(egory) of the main verb of the sentence and a number of other opacity operators, like the presence of future tense, a question or an exclamative, the presence of modals, etc.
- CHANGE IN THE WORLD: to establish whether a clause describes a change or not, and which type of change it describes, the algorithm takes into account four parameters: polarity (i.e. affirmative or negative), domain, tense and aspect of the main verb.

If polarity is set to NO (i.e. if the clause is negative), CHANGE is negated; but if the verb describes a state, CHANGE is null because a stative verb can never express a change, apart from the fact that it is affirmed or negated. Thus, if DISCOURSE DOMAIN is subjective and the verb is stative, CHANGE is null: this captures the fact that, in such a case, the action affects only the subject's mind and has no effects on the outside world. In all other cases the algorithm takes into account tense and aspect of the main verb and obeys the following rules: if tense is simple present, CHANGE is null; if tense is *passato remoto* or simple past, CHANGE is culminated; if tense is pluperfect or *trapassato remoto*, CHANGE is earlier; if tense is the *imperfetto* and describes a state, CHANGE is null, but if it describes an activity, a process, an accomplishment, or if it is a mental activity, CHANGE is gradual.

- FACTIVITY: this relation may only assume two values: factive and non-factive. A factive relation is assigned every time change is non null. Other sources of information may be used to trigger factivity, and that is the presence of a factive predicate, like a presuppositional verb, "know".

We now turn to the cues for direct speech. Once the algorithm has recognized a clause to be in direct speech, the CLAUSE TYPE value is dir_speech/prop. The DISCOURSE DOMAIN is also subjective: this is so because direct speech reports the thoughts and perceptions of the characters in the story, so that any intervention of the writer is left out. As far as CHANGE is concerned, the algorithm obeys the following rules: if the main verb is in the imperative mood, CHANGE is null because, although the imperative is used to express commands, there is no certainty that once a command has been imparted it is going to be carried out. If the verb is in the indicative mood, and it is in the future, CHANGE is null as well since the action has still to take place; if we have a past tense such as the *passato prossimo* or the *trapassato*, CHANGE is culminated or earlier, respectively; if tense is present, the algorithm checks its aspect: if the verb describes a state, CHANGE is null, otherwise (i.e. if the verb describes an activity) CHANGE is gradual. Finally, negative and positive polarity is carefully weighted in case the sentence has a complex structure, taking care of cases of double negations. Positives are so marked when the words searched in the input sentence belong to the class of so-called "Absolute Positives", i.e. words that can only take on positive evaluative meaning. The same applies for Negative polarity words, when they belong to a list of "Absolute Negatives", like swear words.

2. Results and Discussion

Here below is the table of our results for the three tasks of Sentipolc (see Basile et al. 2014).

Task	F-ScoreTot	Prec0	Rec0	F-score0	Prec1	Rec1	F-score1	Rank
Subjectivity	52.24	34.79	30.26	32.37	75.71	68.83	72.11	9th/9
Polarity Pos	51.81	72.97	***81.58***	77.03	43.13	16.05	23.39	10th/11
Polarity Neg	51.81	60.97	***77.00***	68.05	62.03	28.19	38.77	10th/11
Irony	49.29	88.29	77.54	82.57	15.66	16.39	16.02	4th/7

Table 2. Results of ITGetaruns for all Tasks.

In Table 2. we report percent values of our system performance. In a final column we registered our placement in the graded scale of final results. As can be noticed, best result has been achieved for irony detection. In general, we can note the following: there has been always an attempt to favour Recall rather than Precision, and also an attempt to reduce False Positives. This would be represented by a better scoring in those values associated to Prec0, Rec0 and F-score0: as can be noticed, this is only partially true. Both Polarity and Irony have by far better scoring in 0s than in 1s. On

the contrary, Subjectivity has much better scores in 1s than in 0s. We assume that this is due to annotation criteria, which don't match our linguistic rules. We marked with bold italics those scores that have better ranking individually, and both coincide with Recall0 in Polarity. Recall0 for Polarity Pos is 81.58, which corresponds to the 4th rank in the list of 12 (not considering the baseline); Recall0 for Polarity Neg is 77.00, which represents the best result of all systems. Going back to annotation criteria, one of our basic rule for Subjectivity matching is presence of 1st and 2nd person morphology in the main verb complex associated to the main or root clause. We noticed that this does not always coincide with annotations associated to the tweets.

We had a number of additional features to implement, which would have increased Precision quite significantly but would have decreased Recall dramatically. One of these features was the possibility to highlight the use of alterations in Ironic tweets, which are used to express "Exaggeration". The algorithm was based on our Morphological Analyser that in turn is based on linguistic rules for alterations and a root lexicon of Italian made up of some 90,000 entries (see Delmonte, Pianta 1996; 1998). We also intended to use our classification of Emoticons, which however proved not to be a significant contribution in the overall evaluation, so at the end we decided not to implement it. Eventually, we sieved unallowed combinations of 0-1 and replaced the unwanted 1 with a zero.

As a conclusion, we intend to implement those techniques that seemed promising but required deeper inspection and were more time-consuming, like using Emoticons and alterations to detect exaggerations in tweets. This will need to make use of Predicate-Argument Structures in the hope to improve irony detection (but see Reyes & Rosso 2013). By knowing, for instance, that swear words - or exaggerations - are being using in a political context, will constitute a good hint if arguments are properly under control.

References

Basile, V., Bolioli, A., Nissim, M., Patti, V., Rosso, P. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task, Proceedings of EVALITA'14, Pisa.

Bos, Johan & Delmonte, Rodolfo (eds.) 2008. "Semantics in Text Processing (STEP), Research in Computational Semantics", Vol.1, College Publications, London.

Delmonte R., 2009. Computational Linguistic Text Processing - Lexicon, Grammar, Parsing and Anaphora Resolution, Nova Science Publishers, New York.

Delmonte, R. 2007. Computational Linguistic Text Processing – Logical Form, Logical Form, Semantic Interpretation, Discourse Relations and Question Answering, Nova Science Publishers, New York.

Delmonte, R., Tonelli, S., Tripodi, R. 2010. Semantic Processing for Text Entailment with VENSES, published at http://www.nist.gov/tac/publications/ 2009/papers.html in TAC 2009 Proceedings Papers.

Delmonte, R. (2009). Computational Linguistic Text Processing – Lexicon, Grammar, Parsing and Anaphora Resolution, Nova Science Publishers, New York.

Delmonte R. and Vincenzo Pallotta, 2011. Opinion Mining and Sentiment Analysis Need Text Understanding, in "Advances in Distributed Agent-based Retrieval Tools", "Advances in Intelligent and Soft Computing", Springer, 81-96.

Esuli, A. and F. Sebastiani 2006. SentiWordnet: a publicly available lexical resource for opinion mining. In Proceedings of the 5th Conference on Language Resources and Evaluation LREC, 6.

Kim, S.-M. and E. Hovy, 2004. Determining the sentiment of opinions. In Proceedings of the 20th international conference on computational linguistics (COLING 2004), 1367–1373.

Pang, B. and L. Lee, 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL), 271–278.

Polanyi, Livia and Zaenen, Annie 2006. "Contextual valence shifters". In Janyce Wiebe, editor, Computing Attitude and Affect in Text: Theory and Applications. Springer, Dordrecht, 1–10.

Reyes A., Rosso P. 2013. On the Difficulty of Automatically Detecting Irony: Beyond a Simple Case of Negation. In: Knowledge and Information Systems.

Saurì R., Pustejovsky, J., 2012. "Are You Sure That This Happened? Assessing the Factuality Degree of Events in Text", Computational Linguistics, 38, 2, 261-299.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. & Stede, M. 2011. "Lexicon-based methods for sentiment analysis". In Computational Linguistics 37(2): 267-307.

Turney, P.D. and M.L. Littman, 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS), 15–346.

Wiebe, Janyce, Wilson, Theresa, Cardie, Claire 2005. "Annotating expressions of opinions and emotions in language". In Language Resources and Evaluation, 39(2): 165–210.

10.12871/clicit2014212

Subjectivity, Polarity And Irony Detection: A Multi-Layer Approach

Elisabetta Fersini, Enza Messina, Federico Alberto Pozzi
DISCo, University of Milano-Bicocca
Viale Sarca 336
20126 - Milan
{fersini,messina,federico.pozzi}@disco.unimib.it

Abstract

English. In the literature, subjectivity, polarity and irony detection have been often considered as independent tasks. However, since there are multiple ties between them, they should be jointly addressed. In this paper we propose a hierarchical system, where the classifiers of each layer are built upon an ensemble approach known as Bayesian Model Averaging.

Italiano. *In letteratura, le classificazioni di soggettività, polarità e ironia sono state spesso affrontate come task indipendenti. Tuttavia, dal momento che esistono tra loro diversi legami impliciti, tali task dovrebbero essere affrontati congiuntamente. In questo lavoro proponiamo un sistema gerarchico, dove i classificatori di ogni layer sono costruiti ricorrendo ad un approccio di ensemble learning noto come Bayesian Model Averaging.*

1 Introduction

Among the computational approaches for distinguishing subjective vs objective messages, ironic vs not ironic and different classes of polarities, we can point out two main research directions: the first one focuses on machine learning algorithms for automatic recognition (Pang et al., 2002; Chen et al., 2008; Ye et al., 2009; Perea-Ortega et al., 2013; Pozzi et al., 2013c; Pozzi et al., 2013a), while the second one is aimed at the identification of linguistic and metalinguistic features useful for automatic detection (Carvalho et al., 2009; Filatova, 2012; Pozzi et al., 2013b; Davidov et al., 2010; Reyes et al., 2013). As far is concerned with the machine learning perspective, although some approaches are widely used in sentiment analysis, they suffer from two main limitations that the proposed paper intends to overcome. First, all the issues related to sentiment analysis are usually approached by focusing on specific tasks separately, i.e. subjectivity, polarity and irony are tackled independently on each other. In a real context all these issues should be addressed by a single model able to distinguish at first if a message is either subjective or objective, to subsequently address polarity and irony detection and deal with the potential relationships that could exists between them. Second, within the sentiment analysis research field there is no agreement on which machine learning methodology is better than others: one learner could perform better than others in respect of a given application domain, while a further approach could outperform the others when dealing with a given language or linguistic register. In this paper we present a system based on a multi-layer Bayesian ensemble learning that tries to overcome the above mentioned limitations. The focus is therefore intentionally on learning strategies instead of on linguistic aspects to investigate the potential of multiple and interconnected layers of ensembles on real word Italian Twitter data.

2 Description of the system

2.1 Hierarchical Bayesian Model Averaging

In the literature, *subjectivity*, *polarity* and *irony* detection have been often considered as independent tasks. However, since there are multiple ties between them, they should be jointly addressed. Different works have usually treated subjectivity and polarity classification as two-stage binary classification process, where the first level distinguishes subjective and objective (neutral) statements, and the second level then further distinguishes subjectivity into: subjective-positive / subjective-negative (Refaee and Rieser, 2014; Baugh, 2013). The results proposed in (Wilson et

al., 2009) support the validity of this process, indicating that the ability to recognize neutral classes in the first place can greatly improve the performance in distinguishing between positive and negative utterances at a later time. However, as briefly introduced, also irony can give its contribution in improving the classification performance. An ironic message involves a shift in evaluative valence, which can be treated in two ways: it could be a shift from a literally positive to an intended negative meaning, or a shift from a literally negative to an intended positive evaluation.

According to the above mentioned considerations, we propose a hierarchical framework able to jointly address subjectivity, polarity and irony detection. An overview of the working system, named *Hierarchical Bayesian Model Averaging* (H-BMA), is presented in Figure 1.

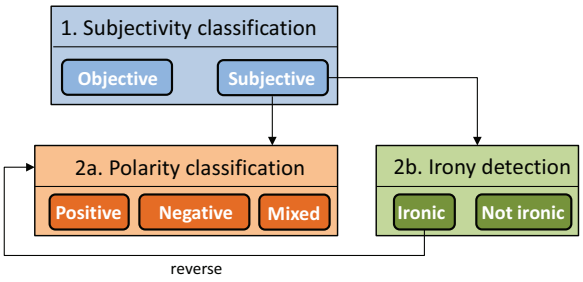


Figure 1: Hierarchical BMA.

Since subjectivity classification is usually the most performing task in Sentiment Analysis, the first level distinguishes subjective and objective statements (neutral is supposed to be objective), and the second level then distinguishes subjectivity into: subjective-positive / subjective-negative / subjective-mixed (a sentence which is subjective, positive and negative at the same time). Jointly with polarity classification, irony detection is also performed. If a given sentence is detected as ironic, then its positive or negative polarity is reversed. On the other side, if the sentence is ironic but its polarity has been classified as mixed, then it is switched to negative. Thus a message s, identified as mixed by the polarity classification layer and ironic (denoted as *iro*) by the irony detection layer, is finally labelled as negative ($-$) due to the conditional distribution

$$P(s = \text{-} \mid s = \text{iro}) >> P(s = \text{+} \mid s = \text{iro}) \quad (1)$$

In the literature, *subjectivity*, *polarity* and *irony* detection have been often addressed applying the most varied machine learning approaches. As outlined in the Introduction, there is no agreement on which methodology is better than others. The uncertainty about which model represents the optimal one in different context has been overcome in this work by introducing Bayesian Model Averaging (Pozzi et al., 2013a), a novel ensemble learning approach able to exploit the potentials of several learners when predicting the labels for each task (subjectivity, irony and polarity) of the hierarchical framework.

2.2 Bayesian Model Averaging

The most important limitation of traditional ensemble approaches is that the models to be included in the *set of experts* have uniform distributed weights regardless their reliability. However, the uncertainty left by data and models can be filtered by considering the Bayesian paradigm. In particular, through Bayesian Model Averaging (BMA) all possible models in the hypothesis space could be used when making predictions, considering their marginal prediction capabilities and their reliability. Given a dataset $\mathcal{D}$ and a set C of classifiers, the approach assigns to a message s the label $l(s)$ that maximizes:

$$P(l(s) \mid C, \mathcal{D}) = \sum_{i \in C} P(l(s) \mid i, \mathcal{D}) P(i \mid \mathcal{D}) \quad (2)$$

where $P(l(s) \mid i, \mathcal{D})$ is the marginal distribution of the label predicted by classifier i and $P(i \mid \mathcal{D})$ denotes the posterior probability of model i. The posterior $P(i \mid \mathcal{D})$ can be computed as:

$$P(i \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid i) P(i)}{\sum_{j \in C} P(\mathcal{D} \mid j) P(j)} \quad (3)$$

where $P(i)$ is the prior probability of i and $P(\mathcal{D} \mid i)$ is the model likelihood. In eq. 3, $P(i)$ and $\sum_{j \in C} P(\mathcal{D} \mid j) P(j)$ are assumed to be a constant and therefore can be omitted. Therefore, BMA assigns the label $l^{BMA}(s)$ to s according to the following decision rule:

$$\begin{aligned} l^{BMA}(s) &= \arg\max_{l(s)} P(l(m)|C, \mathcal{D}) \\ &= \sum_{i \in C} P(l(s)|i, \mathcal{D}) P(i|\mathcal{D}) \\ &= \sum_{i \in C} P(l(s)|i, \mathcal{D}) P(\mathcal{D}|i) P(i) \\ &= \sum_{i \in C} P(l(s)|i, \mathcal{D}) P(\mathcal{D}|i) \end{aligned} \quad (4)$$

We proposed to replace the implicit measure $P(\mathcal{D} \mid i)$ by an explicit estimate, known as F_1-measure, obtained during a preliminary evaluation of the classifier i. In particular, by performing a cross validation, each classifier can produce an average measure stating how well a learning machine generalizes to unseen data. Considering ϕ-folds for cross validating a classifier i, the measure $P(\mathcal{D} \mid i)$ can be approximated as

$$P(\mathcal{D} \mid i) \approx \frac{1}{\iota} \sum_{\iota=1}^{\phi} \frac{2 \times P_{i\iota}(\mathcal{D}) \times R_{i\iota}(\mathcal{D})}{P_{i\iota}(\mathcal{D}) + R_{i\iota}(\mathcal{D})} \quad (5)$$

where $P_{i\iota}(\mathcal{D})$ and $R_{i\iota}(\mathcal{D})$ denotes precision and recall obtained by classifier i in fold ι.

In this way we tune the probabilistic claim of each classifier in the ensemble according to its ability to fit the training data. This approach allows the uncertainty of each classifier to be taken into account, avoiding over-confident inferences.

A crucial issue of most ensemble methods is referred to the selection of the optimal set of models to be included in the ensemble. This is a combinatorial optimization problem over $\sum_{p=1}^{N} \frac{N!}{p!(N-p)!}$ possible solutions where N is the number of classifiers and p represents the dimension of each potential ensemble. Several metrics have been proposed in the literature to evaluate the contribution of classifiers to be included in the ensemble (see (Partalas et al., 2010)). To the best of our knowledge this measures are not suitable for a Bayesian Ensemble, because they assume uniform weight distribution of classifiers. In this study, we used a heuristic able to compute the discriminative marginal contribution that each classifier provides with respect to a given ensemble. In order to illustrate this strategy, consider a simple case with two classifiers named i and j. To evaluate the contribution (gain) that the classifier i gives with respect to j, we need to introduce two cases:

1. j incorrectly labels the sentence s, but i correctly tags it. This is the most important contribution of i to the voting mechanism and represents how much i is able to correct j's predictions;

2. Both i and j correctly label s. In this case, i corroborates the hypothesis provided by j to correctly label the sentence.

On the other hand, i could also bias the prediction in the following cases:

3. j correctly labels sentence s, but i incorrectly tags it. This is the most harmful contribution in a voting mechanism and represents how much i is able to negatively change the (correct) label provided by j.

4. Both i and j incorrectly label s. In this case, i corroborates the hypothesis provided by j leading to a double misclassification of s.

To formally represent the cases above, let compute $P(i = 1 \mid j = 0)$ as the number of instances correctly classified by i over the number of instances incorrectly classified by j (case 1) and $P(i = 1 \mid j = 1)$ the number of instances correctly classified both by i over the number of instances correctly classified by j (case 2). Analogously, let $P(i = 0 \mid j = 1)$ be the number of instances misclassified by i over the number of instances correctly classified by j (case 3) and $P(i = 0 \mid j = 0)$ the number of instances misclassified by i over the number of instances misclassified also by j (case 4).

The contribution r_i^S of each classifier i belonging to a given ensemble $S \subseteq C$ can be estimated as:

$$r_i^S = \frac{\sum\limits_{j \in \{S \setminus i\}} \sum\limits_{q \in \{0,1\}} P(i = 1 \mid j = q) P(j = q)}{\sum\limits_{j \in \{S \setminus i\}} \sum\limits_{q \in \{0,1\}} P(i = 0 \mid j = q) P(j = q)} \quad (6)$$

where $P(j = q)$ is the prior of classifier j to either correctly or incorrectly predict labels. In particular, $P(j = 1)$ denotes the percentage of correctly classified instances (i.e. accuracy), while $P(j = 0)$ represents the rate of misclassified (i.e. error rate).

Once the contribution of each classifier has been computed, a further issue to be addressed concerns with the search strategy for determining the optimal ensemble composition. The proposed evaluation function r_i^S is included in a greedy strategy based on backward elimination: starting from an initial set $S = C$, the contribution r_i^S is iteratively computed excluding at each step the classifier that achieves the lowest r_i^S. The proposed strategy allows us to reduce the search space from $\sum_{p=1}^{n} \frac{n!}{p!(n-p)!}$ to $n-1$ potential candidates for determining the optimal ensemble, because at each step the classifier with the lowest r_i^S is disregarded until the smallest combination is achieved. Another issue that concerns greedy selection is the stop condition related to the search process, i.e.

how many models should be included in the final ensemble. The most common approach is to perform the search until all models have been removed from the ensemble and select the sub-ensemble with the lowest error on the evaluation set. Alternatively, other approaches select a fixed number of models. In this paper, we perform a backward selection until a local maxima of average classifier contribution is achieved. In particular, the backward elimination will continue until the Average Classifier Contribution (ACC) of a sub-ensemble with respect to the parent ensemble will decrease. Indeed, when the average contribution decreases the parent ensemble corresponds to a local maxima and therefore is accepted as optimal ensemble combination. More formally, an ensemble S is accepted as optimal composition if the following condition is satisfied:

$$\frac{ACC(S)}{|S|} \geq \frac{ACC(S \setminus x)}{|S-1|} \tag{7}$$

where $ACC(S)$ is estimated as the average r_i^S over the classifiers belonging to the ensemble S. Note that the contribution of each classifier i is computed according to the ensemble S, that is iteratively updated once the worst classifier is removed. This leads to the definition of S characterized by a decreasing size ranging from $|S| = N, N-1, \ldots, 1$.

3 Results

In order to derive the feature space used for learning, a vector space model has been adopted. Each sentence s is represented as a vector composed of terms for which a corresponding weight w can be computed as Boolean (0/1). No additional information, such as linguistic cues, has been provided to the learning approaches investigated in this paper. The proposed Hierarchical Bayesian Model Averaging (H-BMA) has been compared with traditional Bayesian Model Averaging (BMA) and the baseline provided by Sentipolc 2014 organizers (Basile et al., 2014). The classifiers enclosed in H-BMA and BMA for addressing the three tasks are: Decision Tree (DT) (Quinlan, 1993), Support Vector Machines (SVM) (Vapnik and Vapnik, 1998), Multinomial Naive Bayes (MNB) (Langley et al., 1992) and K-Nearest Neighbors (KNN) (Aha et al., 1991). The indices used for comparing the approaches are Precision, Recall and F_1-measure.

	Baseline	**BMA**	**H-BMA***
Subjectivity	0.4005	0.6173	0.6173
Polarity	0.3718	0.4907	0.5253
Irony	0.4441	0.5253	0.5261

Table 1: Comparison of F_1-measure

The results reported in Table 1 show the F_1-measure performance on the three tasks*. The optimal ensemble composition of both BMA and H-BMA has been obtained according the greedy backward elimination strategy that lead to ensemble composed of DT, SVM and MNB (for all the three tasks). It can be easily noted that addressing Subjectivity, Polarity and Irony detection with H-BMA, where tasks are modelled as interdependent, the performance tend to improve with respect to the other approaches where the issues are tackled independently.

4 Discussion

In this paper, a novel system for jointly modelling subjectivity, polarity and irony detection has been introduced. The experimental results show the potential of the proposed model to address interdependent tasks with no additional information derived by linguistic cues. The proposed solution is particularly effective and efficient, thanks to its ability to define a strategic combination of different classifiers through an accurate and computationally efficient heuristic. However, an increasing number of classifiers to be enclosed in each ensemble in all the layers together with large dataset open to deeper considerations in terms of complexity. The selection of the initial ensemble should consider the different complexities of each single learner and inference algorithm, leading to a reasonable trade-off between their contribution in terms of accuracy and the related computational time. A further ongoing research is related to the linguistic aspects that could be taken into account during the learning phase of the models in the ensembles. Specific linguistic cues able to characterise subjectivity, polarity and irony could lead to more accurate learning and prediction.

*Official results provided to Sentipolc 2014 organizers (Basile et al., 2014) lead to the following F_1-measure performance: Subjectivity 0.5901, Polarity 0.5341 and Irony 0.4771. The results reported in Table 1 differ from the ones reported in the official ranking because of a mistake in sending the correct predictions.

References

David W Aha, Dennis Kibler, and Marc K Albert. 1991. Instance-based learning algorithms. *Machine learning*, 6(1):37–66.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*, Pisa, Italy.

Wesley Baugh. 2013. bwbaugh : Hierarchical sentiment analysis with partial self-training. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 539–542. Association for Computational Linguistics.

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

Bo Chen, Hui He, and Jun Guo. 2008. Constructing maximum entropy language models for movie review subjectivity analysis. *Journal of Computer Science and Technology*, 23(2):231–239.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 107–116, Stroudsburg, PA, USA. Association for Computational Linguistics.

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *LREC*, pages 392–398.

Pat Langley, Wayne Iba, and, and Kevin Thompson. 1992. An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, pages 223–228. AAAI Press.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Ioannis Partalas, Grigorios Tsoumakas, and Ioannis Vlahavas. 2010. An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. *Machine Learning*, 81(3):257–282.

José M Perea-Ortega, Eugenio Martínez-Cámara, María-Teresa Martín-Valdivia, and L Alfonso Ureña-López. 2013. Combining supervised and unsupervised polarity classification for non-english reviews. In *Computational Linguistics and Intelligent Text Processing*, pages 63–74. Springer.

Federico Alberto Pozzi, Elisabetta Fersini, and Enza Messina. 2013a. Bayesian model averaging and model selection for polarity classification. In *Proceedings of the 18th International Conference on Applications of Natural Language to Information Systems*, pages 189–200.

Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Daniele Blanc. 2013b. Enhance polarity classification on social media through sentiment-based feature expansion. In *Proceedings of the 14th Workshop "From Objects to Agents" co-located with the 13th Conference of the Italian Association for Artificial Intelligence (AI*IA 2013), Torino, Italy, December 2-3, 2013.*, pages 78–84.

Federico Alberto Pozzi, Daniele Maccagnola, Elisabetta Fersini, and Enza Messina. 2013c. Enhance user-level sentiment analysis on microblogs with approval relations. In *AI* IA 2013: Advances in Artificial Intelligence*, pages 133–144. Springer.

John Ross Quinlan. 1993. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann.

Eshrag Refaee and Verena Rieser. 2014. Subjectivity and sentiment analysis of arabic twitter feeds with limited resources. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference*, LREC14, pages 16–21.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. 1998. *Statistical learning theory*, volume 2. Wiley New York.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Comput. Linguist.*, 35(3):399–433.

Qiang Ye, Ziqiong Zhang, and Rob Law. 2009. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, 36(3):6527–6535.

10.12871/clicit2014213

IRADABE: Adapting English Lexicons to the Italian Sentiment Polarity Classification task

Irazú Hernández Farias
Pattern Recognition and
Human Language Technology
Universitat Politècnica de València
Spain
dhernandez1@dsic.upv.es

Davide Buscaldi
Laboratoire d'Informatique de Paris Nord
CNRS (UMR 7030)
Université Paris 13, Sorbonne Paris Cité
France
buscaldi@lipn.univ-paris13.fr

Belém Priego Sánchez
Laboratoire de Lexiques, Dictionnaires, Informatique de Paris Nord
CNRS(UMR 7187)
Université Paris 13, Sorbonne Paris Cité
France
belemps@gmail.com

Abstract

English. Interest in the Sentiment Analysis task has been growing in recent years due to the importance of applications that may benefit from such kind of information. In this paper we addressed the polarity classification task of Italian tweets by using a supervised machine learning approach. We developed a set of features and used them in a machine learning system in order to decide if a tweet is subjective or objective. The polarity result itself was then used as an additional feature to determine whether a tweet contains ironical content or not. We faced the lack of resources in Italian by translating (mostly automatically) existing resources for the English language. Our model obtained good results in the SentiPolC 2014 task, being one of the best ranked systems.

Italiano. *L'interesse nell'analisi automatica dei sentimenti è continuamente cresciuto negli ultimi anni per via dell'importanza delle applicazioni in cui questo tipo di analisi può essere utilizzato. In quest'articolo descriviamo gli esperimenti portati a termine nel campo della classificazione di polarità di tweets scritti in italiano, usando un approccio basato sull'apprendimento automatico. Un certo numero di criteri è stato utilizzato come features per assegnare una polarità e quindi determinare se i tweets contengono dell'ironia o meno. Per questi esperimenti, la mancanza di risorse (in particolare di dizionari specializzati) è stata compensata adattando, in gran parte utilizzando delle tecniche di traduzione automatica, delle risorse esistenti per la lingua inglese. Il modello così ottenuto è stato uno dei migliori nel task SentiPolC a Evalita 2014.*

1 Introduction

Sentiment Analysis has been defined by (Liu, 2010) as "the computational study of opinions, sentiments and emotions expressed in text"; social media is a rich source of data that can be processed in order to detect subjectivity and classify the sentiments expressed by users. The effective extraction of such information is the main challenge in this research field. Sentiment analysis is an opportunity for researchers in Natural Language Processing (NLP) to make tangible progress on all fronts of NLP, and potentially have a huge practical impact. (Cambria et al., 2013)

In this paper we describe our participation to the SentiPolC task in polarity and irony classification of tweets in Italian. The paper is organized as follows: in Section 2 we briefly describe the related works in order to understand how they influenced our choices. In Section 3 we describe the features and the classification system used. Results obtained from our proposed model are shown in Section 4. Finally in Section 5 we draw some conclusions based on the early analysis of the results.

2 Related Work

Sentiment Analysis approaches are mainly based on machine learning and lexicons. There is a considerable amount of works related to sentiment analysis and opinion mining ((Liu, 2010), (Pang and Lee, 2008) in particular), all of them can be classified in one of the general approaches presented by Cambria et. al in (Cambria et al., 2013): keyword spotting, lexical affinity, statistical methods, and concept-based techniques. *Keyword spotting* consists in classifying text by affect categories based on the presence of unambiguous affect words such as *happy* , *sad*, *afraid*, and *bored*. *Lexical affinity* does not only detects obvious affect words, but also assigns to arbitrary words a probable "affinity" to particular emotions. *Statistical methods* are semantically weak, which means that individually — with the exception of obvious affect keywords — a statistical model's other lexical or co-occurrence elements have little predictive value. *Concept-based approaches*: relying on large semantic knowledge bases, such approaches step away from blindly using keywords and word co-occurrence counts, and instead rely on the implicit meaning/features associated with natural language concepts, superior to purely syntactical techniques; concept-based approaches can detect subtly expressed sentiments.

Respect to irony detection, Carvalho (Carvalho et al., 2009) developed a system able to detect irony using punctuation marks and emoticons in Portuguese. Veale and Hao (Veale and Hao, 2010) present a linguistic approach that takes into account the presence of heuristic clues in sentences (e.g. "about as" as indicator of ironic simile). Reyes et al. (Reyes et al., 2013) propose a model based on four dimensions (signatures, unexpectedness, style, and emotional scenarios) that support the idea that textual features can capture patterns used in this kind of utterances.

3 Features and Classification Framework

In order to address the tasks of subjectivity/polarity/ironic classification, we decide taking into account a statistical method that includes several features: structural, syntactical and lexicon based. We think that tweets belonging to the same class can share this kind of features, below we describe briefly each one. In parentheses, we provide the related id used in Table 4 and Table 5.

3.1 Surface Features

- *nGrams features.* We extracted the most frequent unigrams, bigrams and trigrams from the training corpus in order to have three different Bag of Words representations. This is a simple feature widely used in text classification. Only unigrams were finally used for our participation in SentiPolC.

- *Emoticons frequency.* (*emo*) By using emoticons, with few characters is possible to display one's true feeling. Emoticons are virtually required under certain circumstances in text-based communication, where the absence of verbal and visual cues can otherwise hide what was originally intended to be humorous, sarcastic, ironic, and some times negative (Wolf, 2000). We manually defined three different sets of emoticons for the detection of subjectivity, positiveness and negativeness, then we extracted the frequency of each one in tweets.

- *Negative Words frequency.* (*neg*) Handling negation can be an important concern in sentiment analysis, one of the main difficulties is that negation can often be expressed in a rather subtle way. We analyzed the training set and selected some words that triggers negation (*mai* (never), *non/no* (not/no)), avversative conjunction or adverbs (*invece* (instead), *ma* (but)). We extracted their frequency in each tweet. There are other ways to deal with negations, for example to reverse the polarity of the text if a negation word is found, but we did not employ this technique.

- *URL information frequency.* (*http*) We analyzed the training set and we found that most not-subjective, not-ironic tweets contained a hyperlink, so we decided to take into account this characteristic as a feature. In some cases this kind of information is also present in ironic tweets because users made an evaluation of some content (text, video, image, etc.) that they consider ironic and try to share with others in order to express themselves.

- *POS-based features.* (*pps*) We decided to use Part-of-speech (POS) tagging (the TreeTagger[1] implementation) to extract additional in-

[1] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

formation to determine the subjectivity of tweets; in particular, we took into account the presence of verbs conjugated at the first and second persons (those endings in "-o", "-i", "-amo", "-ate/ete") and personal pronouns ("io", "tu", "noi", "voi", and their direct and indirect object versions).

- *Tweet Length and Uppercase ratio.* (*len*, *shout*) Although text in tweets only can contain maximum 140 characters, we decided to use the length in words of each tweet like a feature, trying to reflect the fact that ironic comments are often short. We took into account also the ratio between the uppercase words and length of the tweet, given that many subjective and/or ironic comments use uppercase words in order to express radical opinions about something, highlighting it with the use of uppercase.

3.2 Lexicon-based Features

Many state-of-the-art works are based on lexicons that assign to each words an empirical measure of their polarity. Most lexicons however are available only in English. We decided to use different lexicons and automatically translate them to Italian; a thoroughful description of each one is out of the scope of the present work and we refer the reader to the relative existing literature. We found that in some cases an Italian word can be translated in different ways in English. We tested on the dev set two possibilities: to keep for the Italian word the max of the scores of the English translations or their average. The test showed that the max allowed to obtain a slightly better accuracy than the average.

- *SentiWordNet* (*SWN*). Assigns to each synset of WordNet three sentiment scores: positivity, negativity and objectivity. We used only the positive and negative scores to derive six features: positive/negative words count (*SWN+/-c*), the sum of the positive scores in the tweet (*SWN+s*), the sum of negative scores in the tweet (*SWN-s*), the balance (positive-negative) score of the tweet (*SWNb*), and the standard deviation of SentiWN scores in the tweet (*SWNdev*).

- *Hu-Liu Lexicon*[2]. (*HL*) We derived three features from this lexicon: positive (*HL+c*) and negative (*HL-c*) words count, balance (sum of positive-negative words - *HLb*).

- *AFINN Lexicon*[3]. (*AF*) This lexicon contains two word lists labeled with polarity valences from -5 (negative) to +5 (positive). We derived 5 features from this lexicon: positive/negative word count (*AF+/-c*), sum of positive and negative scores (*AF+/-s*); overall balance of scores in the tweet (*AFb*).

- *Whissel Dictionary* (Whissell, 2009). (*WH*) Our translation of this lexicon contains 8700 Italian words with values of Activation, Imagery and Pleasantness related to each one. Range of scores go from 1 (most passive) to 3 (most active). We derived six features: average activation, imagery and pleasantness (*WH[aip]avg*), and the standard deviation of the respective scores (*WH[aip]dev*). We thought that an elevate score in one of these features may indicate an out-of-context word, thus indicating a possibly ironic comment.

- *Italian "Taboo Words"*. (*TAB*) Knowing the function of taboo words to trigger humor, catharsis, or to boost opinions (Zhou, 2010), we decided to use a list of taboo italian words that we extracted from Wiktionary[4].

- *Counter-Factuality* (Reyes et al., 2013). (*CF*) We use the frequency of discursive terms that hint at opposition or contradiction in a text such as "about" and "nevertheless".

- *Temporal Compression* (Reyes et al., 2013). (*TC*) We use the frequency of terms that identify elements related to opposition in time, i.e. terms that indicate an abrupt change in a narrative.

Moreover, in the irony subtask we used as features our results of the subjectivity (*subj*) and polarity (*pol*) classification subtasks.

3.3 Classification Framework

We used the nu-SVM (Schölkopf et al., 2000) implementation by LibSVM (Chang and Lin, 2011),

[2] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[3] https://github.com/abromberg/sentiment_analysis/blob/master/AFINN/AFINN-111.txt

[4] http://it.wiktionary.org/wiki/Categoria:Parole_volgari-IT

with the nu parameter set to the standard value (0.5), with a RBF kernel. The classification was carried out in three steps: in the first one, the system classifies the tweet into subjective or not. The result of the subjectivity is passed as a feature to the second classification step that classifies the tweets as positive or negative. Finally, the results of subjectivity and polarity classification are passed to the final classifier that is used to detect irony. In the constrained run, we used the full SentiPolC training set (Basile et al., 2014). In the unconstrained run, we integrated into the training set 493 additional tweets that include the hashtag *#ironia* or were published on an ironical/satirical account (for instance, the *@spinozait* account[5]). We randomly subsampled the training set in order to obtain a balanced training set (with $50\%/50\%$ ratio for the ironic/not ironic tweets).

The additional tweets retrieved from *@spinozait* and those including the hashtag *#ironia* were automatically assigned the labels "1" for subjectivity and irony. The labels for polarity were automatically assigned using the model trained on the devset. This means that in some cases the combination of labels does not correspond to the labels allowed by the task guidelines (there are ironic tweets with mixed or neutral polarity). Therefore, we did not use the polarity information as feature for the unconstrained run.

4 Results

We evaluated our approach on the SentiPolC datasets, composed by approximately 4,000 italian tweets for training and 1,700 for test; each tweet on the training subset was labeled as objective/subjective, positive/neutral/negative/mixed, ironic/non-ironic and finally if the topic of the tweet was concern to politics. In Table 4 we show the results obtained on the training set using 10-fold cross validation. The official results are shown in Table 4 (Basile et al., 2014). The differences between the results obtained for the training and the test set can be explained by the fact that our system was not able to retrieve 186 tweets. Our evaluation on Weka on the partial set shows 80% F-measure in irony detection. However, we suppose that the other participants had similar problems. The results in Table 4 have been calculated only on the retrieved tweets of the training set.

[5] https://twitter.com/spinozait

	Subj	***Pol(+)***	***Pol(-)***	***Iro***
Precision	0.765	0.767	0.668	0.820
Recall	0.777	0.774	0.670	0.828
F-Measure	0.764	0.743	0.668	0.824

Table 1: Results of our model on training set

		Constrained			
		Subj	***Pol(+)***	***Pol(-)***	***Iro***
"1"	P	0.8284	0.7265	0.6822	0.2400
	R	0.7862	0.2998	0.5213	0.2521
	F-m	0.8067	0.4245	0.5910	0.2459
Comb F-m		0.6706	0.6347		0.5415

Table 2: Results of our model on test set Constrained Run (official results).

We carried out an analysis of the features using the information gain feature selection algorithm provided by Weka. We show in Table 4 and Table 5 the ten best dictionary-based features, in the test and training set respectively.

From these results we can see that SentiWordNet-based features worked very well in subjectivity prediction, more than features like the emoticons which we expected to have an important role. In the positive polarity task, emoticons were an important feature however, together with the positive word counts (or sum of positive scores) for AFINN, Hu-Liu and SentiWordNet lexicons. The respective negative word based features worked well also in the negative polarity prediction task. In the irony task we observed some discrepancies between the results obtained in the training set and those obtained in the test set. In fact, our intuition of finding "anomalies" using standard deviation of Whissell-based features worked particularly well in the training set, but we did not found the same kind of "anomalies" in the test set. In the test set we found instead a prevalence of features that

		Unconstrained			
		Subj	***Pol(+)***	***Pol(-)***	***Iro***
"1"	P	0.8955	0.4565	0.6266	0.2387
	R	0.5989	0.5556	0.5040	0.4202
	F-m	0.7178	0.5012	0.5587	0.3044
Comb F-m		0.6464	0.6108		0.5513

Table 3: Results of our model on test set Unconstrained Run(official results).

	Subj	*Pol(+)*	*Pol(-)*	*Iro*
1	*http*	*SWNb*	*SWN-s*	*subj*
2	*SWN+c*	*AFb*	*SWN-c*	*http*
3	*SWN-s*	*emo*	*HL-c*	*HL-c*
4	*SWN+s*	*AF+s*	*AF-s*	*pol*
5	*SWN-c*	*HLb*	*SWNb*	*AF-c*
6	*SWNdev*	*SWN+s*	*HLb*	*HLb*
7	*AFb*	*AF+c*	*AF-c*	*SWN-s*
8	*neg*	*WHidev*	*neg*	*AFb*
9	*AF+s*	*HL+c*	*CF*	*AF-s*
10	*pps*	*WHpdev*	*AFb*	*SWNb*

Table 4: Best ranked dictionary-based features for each subtask, according to their information gain values (test set).

	Subj	*Pol(+)*	*Pol(-)*	*Iro*
1	*http*	*AFb*	*SWN-s*	*subj*
2	*SWN+c*	*AF+s*	*AF-s*	*http*
3	*SWN+s*	*SWNb*	*HL-c*	*pol*
4	*SWNdev*	*emo*	*SWN-c*	*WHpdev*
5	*SWN-c*	*SWN+s*	*AF-c*	*WHadev*
6	*SWN-s*	*HLb*	*SWNb*	*WHidev*
7	*AFb*	*AF+c*	*AFb*	*len*
8	*SWNb*	*HL+c*	*SWNdev*	*SWN+c*
9	*AF+s*	*http*	*SWN+c*	*SWN-c*
10	*shout*	*len*	*HLb*	*TAB*

Table 5: Best ranked dictionary-based features for each subtask, according to their information gain values (training set).

indicates negative words (*HL-c*, *AF-c*, *SWN-s*, *AF-s*). In both train and test set we observed that the most important features that characterize irony were subjectivity and mixed polarity, while the presence of web addresses was a strong clue to the tweet being not ironic, or objective. The importance of web related features was indicated also by the high information gain of fragments of web addresses (not included in the tables), such as "http", "ly", "it", "fb", etc. Further analysis of the results showed that Italian politics have a great weight in the training set, with keywords like "governo" or "Monti" conveying a high predictive power.

5 Conclusions and Future Work

An analysis of the features using information gain showed that SentiWordNet was an important resource for the detection of subjectivity, and in general the translated lexicons were very useful. Many of the features related to the detection of web addresses were also very important, indicating that the training and test sets were flawed by the presence of such addresses. Finally, we noticed that the lexicon-based features using standard deviation performed particularly well on the irony detection task, at least in the training set, indicating that our intuition of finding "anomalies" was right. We plan to work furtherly in this direction as to detect anomalies in content or changes in polarity from one fragment of text to another and integrate them as further features.

Acknowledgments.

This work is partially supported by a public grant overseen by the French National Research Agency (ANR) as part of the progam "Investissements d'Avenir" (reference: ANR-10-LABX-0083). The National Council for Science and Technology (CONACyT-Mexico) has funded the research work of the first author (218109/313683 grant).

References

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*, Pisa, Italy.

Erick Cambria, B. Schuller, Yunqing Xia, and C. Havasi. 2013. New avenues in opinion mining and sentiment analysis. *Intelligent Systems, IEEE*, 28(2):15–21, March.

Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: Oh...!! it's "so easy" ;-). In *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, pages 53–56, New York, NY, USA. ACM.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268.

Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. 2000. New support vector algorithms. *Neural computation*, 12(5):1207–1245.

Tony Veale and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. *Frontiers in Artificial Intelligence and Applications: ECAI*, 215:765–770.

Cynthia Whissell. 2009. Using the revised dictionary of affect in language to quantify the emotional undertones of samples of natural language 1, 2. *Psychological reports*, 105(2):509–521.

Alecia Wolf. 2000. Emotional expression online: Gender differences in emoticon use. In *CyberPsychology & Behavior*, volume 3.

Ningjue Zhou. 2010. Taboo language on the internet : An analysis of gender differences in using taboo language.

10.12871/clicit2014214

Linguistically–motivated and Lexicon Features for Sentiment Analysis of Italian Tweets

Andrea Cimino[◇], **Stefano Cresci**[•], **Felice Dell'Orletta**[◇], **Maurizio Tesconi**[•]
[◇]Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR)
ItaliaNLP Lab - *www.italianlp.it*
[•]Institute for Informatics and Telematics (IIT–CNR)
{andrea.cimino,felice.dellorletta}@ilc.cnr.it
{stefano.cresci,maurizio.tesconi}@iit.cnr.it

Abstract

English. In this paper we describe our approach to EVALITA 2014 SENTIment POLarity Classification (SENTIPOLC) task. We participated only in the Polarity Classification sub–task. By resorting to a wide set of general–purpose features qualifying the lexical and grammatical structure of a text, automatically created ad–hoc lexicons and existing free available resources, we achieved the second best accuracy[1].

Italiano. *In questo articolo descriviamo il nostro sistema utilizzato per affrontare il compito di Polarity Classification del task SENTIPOLC della conferenza Evalita 2014. Sfruttando un gran numero di caratteristiche generiche che descrivono la struttura lessicale e sintattica del testo, la creazione automatica di lessici ad–hoc e l'uso di risorse disponibili esistenti, il sistema ha ottenuto il secondo miglior punteggio della competizione.*

1 Description of the system

Our approach to the Twitter Sentiment polarity detection task was implemented in a software prototype, i.e. a classifier operating on morpho-syntactically tagged and dependency parsed texts which assigns to each document a score expressing its probability of belonging to a given polarity class. The highest score represents the most probable class. Given a set of features and a training corpus, the classifier creates a statistical model using the feature statistics extracted from the training corpus. This model is used in the classification of unseen documents. The set of features and the machine learning algorithm can be parameterized through a configuration file. For this work, we used linear Support Vector Machines (SVM) using LIBSVM (Chang et al., 2001) as machine learning algorithm.

Since our approach relies on multi–level linguistic analysis, both training and test data were automatically morpho-syntactically tagged by the POS tagger described in (Dell'Orletta, 2009) and dependency-parsed by the DeSR parser using Multi-Layer Perceptron as learning algorithm (Attardi et al., 2009), a state-of-the-art linear-time Shift-Reduce dependency parser.

1.1 Lexicons

In order to improve the overall accuracy of our system, we developed and used sentiment polarity and similarity lexicons. All the created lexicons are made freely available at the following website: *http://www.italianlp.it/software/*.

1.1.1 Sentiment Polarity Lexicons

Sentiment polarity lexicons provide mappings between a word and its sentiment polarity (positive, negative, neutral). For our experiments, we used a publicly available lexicons for Italian and two English lexicons that we automatically translated. In addition, we adopted an unsupervised method to automatically create a lexicon specific for the Italian twitter language.

Existing Sentiment Polarity Lexicons

We used the Italian sentiment polarity lexicon (hereafter referred to as $OPENER$) (Maks et al., 2013) developed within the OpeNER European project[2]. This is a freely available lexicon for the Italian language[3] and includes 24,000 Italian word

[1]Because of an error of the conversion script from our internal format (of the output system) to the official one, we submitted the correct output after the task deadline, as soon as we noticed the error.

[2]http://www.opener-project.eu/

[3]https://github.com/opener-project/public-sentiment-lexicons

entries. It was automatically created using a propagation algorithm and manually reviewed for the most frequent words.

Automatically translated Sentiment Polarity Lexicons

- The Multi–Perspective Question Answering (hereafter referred to as $MPQA$) Subjectivity Lexicon (Wilson et al., 2005). This lexicon consists of approximately 8,200 English words with their associated polarity. In order to use this resource for the Italian language, we translated all the entries through the Yandex translation service[4].

- The Bing Lui Lexicon (hereafter referred to as BL) (Hu et al., 2004). This lexicon includes approximately 6,000 English words with their associated polarity. Like in the former case, this resource was automatically translated by the Yandex translation service.

Automatically created Sentiment Polarity Lexicons

We built a corpus of positive and negative tweets following the Mohammad et al. (2013) approach adopted in the Semeval 2013 sentiment polarity detection task. For this purpose we queried the Twitter API with a set of hashtag seeds that indicate positive and negative sentiment polarity. We selected 200 positive word seeds (e.g. "vincere" *to win*, "splendido" *splendid*, "affascinante" *fascinating*), and 200 negative word seeds (e.g., "tradire" *betray*, "morire" *die*). These terms were chosen from the OPENER lexicon. The resulting corpus is made up of 683,811 tweets extracted with positive seeds and 1,079,070 tweets extracted with negative seeds.

The main purpose of this procedure was to assign a polarity score to each n-gram occurring in the corpus. For each n-gram (we considered up to five n-grams) we calculated the corresponding sentiment polarity score with the following scoring function: $score(ng) = PMI(ng, pos) - PMI(ng, neg)$, where PMI stands for pointwise mutual information. A positive or negative score indicates that the n-gram is relevant for the identification of positive or negative tweets.

1.1.2 Word Similarity Lexicons

Since the lexical information in tweets can be very sparse, to overcame this problem we built two similarity lexicons.

For this purpose, we trained two predict models using the word2vec[5] toolkit (Mikolov et al., 2013). As recommended in (Mikolov et al., 2013), we used the CBOW model that learns to predict the word in the middle of a symmetric window based on the sum of the vector representations of the words in the window. For our experiments, we considered a context window of 5 words. These models learn lower-dimensional word embeddings. Embeddings are represented by a set of latent (hidden) variables, and each word is a multidimensional vector that represent a specific instantiation of these variables. We built the word similarity lexicons by applying the cosine similarity function between the embedded words.

Starting from two corpora, we developed two different similarity lexicons:

- The first lexicon was built using the lemmatized version of the PAISÀ[6] corpus (Lyding et al., 2014). PAISÀ is a freely available large corpus of authentic contemporary Italian texts from the web, and contains approximately 388,000 documents for a total of about 250 millions of tokens.

- The second lexicon was built from a lemmatized corpus of tweets. This corpus was collected starting from 30 generic seed keywords used to query Twitter APIs. The resulting corpus is made up of 1,200,000 tweets. These tweets were automatically morpho-syntactically tagged and lemmatized by the POS tagger described in (Dell'Orletta, 2009).

1.2 Features

In this study, we focused on a wide set of features ranging across different levels of linguistic description. The whole set of features we started with is described below, organised into four main categories: namely, *raw and lexical text features*, *morpho-syntactic features*, *syntactic features* and *lexicon features*. This proposed four–fold partition closely follows the different levels of linguistic analysis automatically carried out on the text being evaluated, (i.e. tokenization, lemmatization, morpho-syntactic tagging and dependency parsing) and the use of external lexical resources.

In the descriptions below, in brackets are reported the names of the features listed in Table 1.

[4] http://api.yandex.com/translate/

[5] http://code.google.com/p/word2vec/

[6] http://www.corpusitaliano.it/

The second column of the table reports for each features the sizes of the used n–grams (for the n–gram features) or it marks whether the considered feature has been used in the final experiment (for the non n–gram features).

1.2.1 Raw and Lexical Text Features

Number of tokens: number of blocks consisting of 5 tokens occurring in the analyzed tweet. *(AVERAGE_TWEET_LENGTH)*

Character n-grams: presence or absence of contiguous sequences of characters in the analyzed tweet. *(NGRAMS_CHARS)*

Word n-grams: presence or absence of contiguous sequences of tokens in the analyzed tweet. *(NGRAMS_WORDS)*

Lemma n-grams: presence or absence of contiguous sequences of lemma occurring in the analyzed tweet. *(NGRAMS_LEMMAS)*

Repetition of n-grams chars: this feature checks the presence or absence of contiguous repetition of characters in the analyzed tweet. *(HAS_NGRAMS_CHARS_REPETITIONS)*

@ Number: number of @ occurring in the analyzed tweet. *(NUM_AT)*

Hashtags number: number of hashtags occurring in the analyzed tweet. *(NUM_HASHTAGS)*

Punctuation: checks whether the analyzed tweet finishes with one of the following punctuation characters: "?", "!". *(FINISHES_WITH_PUNCTUATION)*

1.2.2 Morpho-syntactic Features

Coarse grained Part-Of-Speech n-grams: presence or absence of contiguous sequences of coarse–grained PoS, corresponding to the main grammatical categories (e.g. noun, verb, adjective). *(NGRAMS_CPOS)*

Fine grained Part-Of-Speech n-grams: presence or absence of contiguous sequences of fine-grained PoS, which represent subdivisions of the coarse-grained tags (e.g. the class of nouns is subdivided into proper vs common nouns, verbs into main verbs, gerund forms, past particles). *(NGRAMS_POS)*

Coarse grained Part-Of-Speech distribution: the distribution of nouns, adjectives, adverbs, numbers in the tweet. *(CPOS_DISTR_PERC)*

1.2.3 Syntactic Features

Dependency types n-grams: presence or absence of sequences of dependency types in the analyzed tweet. The dependencies are calculated with respect to *i)* the hierarchical parse tree structure and *ii)* the surface linear ordering of words. *(NGRAMS_DEPTREE, NGRAMS_DEP)*

Lexical Dependency n-grams: presence or absence of sequences of lemmas calculated with respect to the hierarchical parse tree. *(NGRAMS_LEMMATREE)*

Lexical Dependency Triplet n-grams: distribution of lexical dependency triplets, where a triplet represents a dependency relation as (ld, lh, t), where ld is the lemma of the dependent, lh is the lemma of the syntactic head and t is the relation type linking the two. *(NGRAMS_LEMMA_DEP_TREE)*

Coarse Grained Part-Of-Speech Dependency n-grams: presence or absence of sequences of coarse-grained part–of–speech calculated with respect to the hierarchical parse tree. *(NGRAMS_CPOSTREE)*

Coarse Grained Part-Of-Speech Dependency Triplet n-grams: distribution of coarse-grained part–of–speech dependency triplets, where a triplet represents a dependency relation as (cd, ch, t), where cd is the coarse-grained part–of–speech of the dependent, h is the coarse-grained part–of–speech of the syntactic head and t is the relation type linking the two. *(NGRAMS_CPOS_DEP_TREE)*

1.2.4 Lexicon features

Emoticons: presence or absence of positive or negative emoticons in the analyzed tweet. The lexicon of emoticons was extracted from the site *http://it.wikipedia.org/wiki/Emoticon* and manually classified. *(SNT_EMOTICONS)*

Lemma sentiment polarity n-grams: for each lemma n-grams extracted from the analyzed tweet, the feature checks the polarity of each component lemma in the existing sentiment polarity lexicons. Lemma that are not present are marked with the *ABSENT* tag. This is for example the case of the trigram "tutto molto bello" (*all very nice*) that is marked as "*ABSENT-POS-POS*" because *molto* and *bello* are marked as positive in the considered polarity lexicon and *tutto* is absent. The feature is computed for each existing sentiment polarity lexicons. *(NGRAMS_SNT_OPENER, NGRAMS_SNT_MPQA, NGRAMS_SNT_BL)*.

Polarity modifier: for each lemma in the tweet occurring in the existing sentiment polarity lexicons, the feature checks the presence of adjectives or adverbs in a left context window of size 2.

If this is the case, the polarity of the lemma is assigned to the modifier. This is for example the case of the bigram "non interessante" (*not interesting*), where "interessante" is a positive word, and "non" is an adverb. Accordingly, the feature "non_POS" is created. The feature is computed 3 times, checking all the existing sentiment polarity lexicons. *(SNT_WITH_MODIFIER_OPENER, SNT_WITH_MODIFIER_MPQA, SNT_WITH_MODIFIER_BL)*

PMI score: for each set of unigrams, bigrams, trigrams, four-grams and five-grams that occur in the analyzed tweet, the feature computes the score given by $\sum_{\text{i}-gram \in tweet} score(\text{i}-gram)$ and returns the minimum and the maximum values of the five values (approximated to the nearest integer). *(PMI_SCORE)*

Distribution of sentiment polarity: this feature computes the percentage of positive, negative and neutral lemmas that occur in the tweet. To overcome the sparsity problem, the percentages are rounded to the nearest multiple of 5. The feature is computed for each existing lexicon. *(SNT_DISTRIBUTION_OPENER, SNT_DISTRIBUTION_MPQA, SNT_DISTRIBUTION_BL)*

Most frequent sentiment polarity: the feature returns the most frequent sentiment polarity of the lemmas in the analyzed tweet. The feature is computed for each existing lexicon. *(SNT_MAJORITY_OPENER, SNT_MAJORITY_MPQA, SNT_MAJORITY_BL)*

Word similarity: for each lemma of the analyzed tweet, the feature extracts the first 15 similar words occurring in the similarity lexicons. For each similar lemma, the feature checks the presence of negative or positive polarity. In addition, the feature calculates the most frequent polarity. Since we have two different similarity lexicons and three different sentiment lexicons, the feature is computed 6 times. *(COS_EXPLOSION_OPENER_PAISA, COS_EXPLOSION_OPENER_TWITTER, COS_EXPLOSION_MPQA_PAISA, COS_EXPLOSION_MPQA_TWITTER, COS_EXPLOSION_BL_PAISA, COS_EXPLOSION_BL_TWITTER)*

Sentiment polarity in tweet sections: the feature first splits the tweet in three equal sections. For each section the most frequent polarity is computed using the available sentiment polarity lexicons. The purpose of this feature is aimed at identifying change of polarity within the same tweet. *(SNT_POSITION_PRESENCE_OPENER, SNT_POSITION_PRESENCE_MPQA, SNT_POSITION_PRESENCE_BL)*

1.3 Feature Selection Process

Since our approach to Twitter Sentiment polarity detection task relies on a wide number of general-purpose features, a feature selection process was necessary in order to prune irrelevant and redundant features which could negatively affect the classification results. This feature selection process is a variant of the selection method described in (Cimino et al., 2013) used for the Native Language Identification shared task. This new approach has shown better results in terms of the accuracy of the resulting system.

The selection process starts taking into account all the n features described in Section 1.2 and listed in Table 1. The feature selection algorithm drops and adds features until a termination condition is satisfied.

Let F_e be a set containing all the features, and F_d another set of features, initially empty. Let $F_{we} = F_e$ and $F_{wd} = F_d$ two auxiliary sets. In the drop–feature stage, for each feature $f_i \in F_{we}$ we generate a configuration c_i such that the features in $\{f_i\} \cup F_{wd}$ are disabled and all the other features are enabled. When an iteration finishes, we obtain for each c_i a corresponding accuracy score $score(c_i)$ which is computed as as the average of the accuracy obtained by the classifier on five non overlapping test-sets, each one corresponding to the 20% of the training set. We used this five cross fold validation in order to reduce overfitting.

Being c_b the best configuration among all the c_i configurations, and c_B the best configuration found in the previous iterations, if

$$score(c_b) \geq score(c_B) \tag{1}$$

- Move f_b from F_{we} to F_{wd};
- set $F_d := F_{wd}$ and $F_e := F_{we}$;
- set $c_B := c_b$.

If the condition (1) is not satisfied and:

$$score(c_b) + k \geq score(c_B) : \tag{2}$$

- Move f_b from F_{we} to F_{wd}.

For our experiments we set the k initial value to 1.

If the condition (1) or (2) is satisfied, the feature selection process continues with another drop–iteration, otherwise set $k = \frac{k}{2}$.

If $k \leq \alpha$ the feature selection process stops and the configuration c_B is the result of our feature selection process[7]. Otherwise:

- set $F_{wd} := F_d$ and $F_{we} := F_e$,

and the feature selection process continues with the add–feature stage.

In the add–stage we add to the currently best model (c_B) the features previously pruned. For each feature $f_i \in F_{wd}$ we generate a configuration c_i such that the features in $\{f_i\} \cup F_{we}$ are enabled and all the other features are disabled.

For each add–iteration, the process checks the conditions (1) and (2). If the condition (2) is verified and $k \geq \alpha$, another drop–feature stage starts.

In spite of the fact that the described selection process does not guarantee to obtain the global optimum, it however permitted us to obtain an improvement of 2 percentage points (on the five cross validation set) with respect to the starting model indiscriminately using all features.

Table 1 lists the features resulting from the feature selection process.

2 Results and Discussion

Table 2 reports the overall accuracies achieved by our classifier using different feature configuration models in the Polarity Classification task on the official test set. The accuracy is calculated as the average F–score of our system obtained using the evaluation tool provided by the organizers (Basile et al., 2014). Since the official scoring function assigns a bonus also for partial matching (e.g. a Positive or Negative assignment instead of Positive–Negative class), we also report the F–score for each considered polarity class considering only the correct assignments. The first row of the Table shows the results for the *FeatSelLexicons* model resulting from the feature selection process described in section 1.3. This is our official result submitted for the competition. The second row reports the results for the model that uses the same features of the *FeatSelLexicons* classifier where all the lexicon features are disabled. The last row shows the results for the model that contains all the features listed in Table 1. Table 3 reports the

[7]For our experiments we set α to 0.25

Lexical features	
Feature name	n-grams
HAS_NGRAMS_CHARS_REPETITIONS	1 2 **3** **4**
NGRAMS_CHARS	**1** **2** **3** **4**
NGRAMS_WORDS	**1** **2** **3** **4**
NGRAMS_LEMMAS	**1** **2** **3** 4
Feature name	boolean
FINISHES_WITH_PUNCTUATION	True
NUM_AT	True
NUM_HASHTAGS	False
AVERAGE_TWEET_LENGTH	True
SNT_EMOTICONS	True
Morpho–syntactic features	
Feature name	n-grams
NGRAMS_CPOS	1 2 **3**
NGRAMS_POS	1 2 **3**
Feature name	boolean
CPOS_DISTR_PERC	True
Syntactic features	
Feature name	n-grams
NGRAMS_DEP	**1** 2 3
NGRAMS_DEPTREE	**1** **2** 3 4
NGRAMS_LEMMATREE	1 **2** **3** 4
NGRAMS_LEMMA_DEP_TREE	**1** 2 3 4
NGRAMS_CPOSTREE	**1** 2 **3** **4**
NGRAMS_CPOS_DEP_TREE	**1** 2 3 4
Lexicon features	
Feature name	n-grams
NGRAMS_SNT_OPENER	**1** 2 3 **4**
NGRAMS_SNT_MPQA	1 2 3 4
NGRAMS_SNT_BL	**1** **2** **3** 4
NGRAMS_SNT_WITH_MODIFIER_MPQA	**1** **2** 3 4
NGRAMS_SNT_WITH_MODIFIER_BL	**1** 2 **3** 4
Feature name	boolean
COS_EXPLOSION_OPENER_PAISA	True
COS_EXPLOSION_OPENER_TWITTER	True
COS_EXPLOSION_MPQA_PAISA	True
COS_EXPLOSION_MPQA_TWITTER	True
COS_EXPLOSION_BL_PAISA	True
COS_EXPLOSION_BL_TWITTER	False
PMI_SCORE	True
SNT_DISTRIBUTION_OPENER	True
SNT_DISTRIBUTION_MPQA	True
SNT_MAJORITY_OPENER	False
SNT_MAJORITY_MPQA	True
SNT_MAJORITY_BL	False
SNT_POSITION_PRESENCE_OPENER	True
SNT_POSITION_PRESENCE_MPQA	True
SNT_POSITION_PRESENCE_BL	False

Table 1: All the features used for the global model. The features resulting from the features selection process are marked in bold or with the *True* label.

accuracy over the training data before and after the feature selection process. In both cases, we performed a five-fold cross validation evaluation.

For what concerns the results on the official test set, the *AllFeat* model performs slightly better than the *FeatSelLexicons* model, even if the difference in terms of accuracy is not statistically significant. This demonstrates that the lexical, morpho-syntactic, syntactic and lexicon features excluded

Model	Avg. F–score	NEU	POS	NEG	POS_NEG
FeatSelLexicons	0.663	57.1	55.0	62.5	15.3
FeatSelNoLexicons	0.647	56.9	51.0	61.7	11.8
AllFeat	**0.667**	**58.4**	**56.3**	**63.4**	**16.4**

Table 2: Classification results of different feature models on official test data with respect to the four considered classes: Neutral (NEU), Positive (POS), Negative (NEG) and Positive-Negative (POS_NEG).

Model	Avg. F–score
FeatSelLexicons	**0.698**
AllFeat	0.678

Table 3: Classification results obtained by the five-fold cross validation evaluation before and after the feature selection (over the training set).

by the features selection process are not so relevant for this task. The results obtained by the *FeatSelLexicons* classifier show that lexicon features contribute (+1.6 points) to significantly improve the accuracy of our classifier.

3 Conclusion

In this paper, we reported the results of our participation to the Polarity Classification shared task. By resorting to a wide set of general–purpose features qualifying the lexical and grammatical structure of a text and ad hoc created lexicons, we achieved the second best score in the competition.

Current directions of research include adding to our models contextual features derived from contextual information of tweets (e.g. the user attitude, the overall set of recent tweets about a topic), successfully tested by (Croce et al., 2014).

References

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi and Joseph Turian. 2009. Accurate dependency parsing with a stacked multilayer perceptron. In *Proceedings of Evalita '09, Evaluation of NLP and Speech Tools for Italian*. December, Reggio Emilia.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*. December, Pisa.

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a library for support vector machines *Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm*.

Andrea Cimino, Felice Dell'Orletta, Giulia Venturi and Simonetta Montemagni. 2013. Linguistic Profiling based on General–purpose Features and Native Language Identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Application*, 207–215. Atlanta, Georgia. ACL.

Felice Dell'Orletta. 2009. Ensemble system for Part-of-Speech tagging. In *Proceedings of Evalita '09, Evaluation of NLP and Speech Tools for Italian*. December, Reggio Emilia.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04*. 368-177, New York, NY, USA. ACM.

Verena Lyding, Egon Stemle, Claudia Borghetti, Macro Brunello, Sara Castagnoli, Felice Dell'Orletta, Henrik Dittmann, Alessandro Lenci and Vito Pirrelli. 2013. The *PAISÀ* Corpus of Italian Web Texts. In *Proceedings of 9th workshop on Web as Corpus (WAC-9)*. 26 April, Gothenburg, Sweden.

Isa Maks, Ruben Izquierdo, Francesca Frontini, Montse Cuadros, Rodrigo Agerri and Piek Vossen. 2014. Generating Polarity Lexicons with WordNet propagation in 5 languages. *9th LREC, Language Resources and Evaluation Conference*. Reykjavik, Iceland.

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv1:1301.3781.

Saif Mohammad, Svetlana Kiritchenko and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Seventh international workshop on Semantic Evaluation Exercises, SemEval-2013*. 321-327, Atlanta, Georgia, USA.

Andrea Vanzo, Danilo Croce and Roberto Basili. 2014. A context-based model for Sentiment Analysis in Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. August, Dublin, Ireland.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov and Alan Ritter. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05*. 347-354, Stroudsburg, PA, USA. ACL.

10.12871/clicit2014215

The CoLing Lab system for Sentiment Polarity Classification of tweets

Lucia C. Passaro[♣], Gianluca E. Lebani[♣], Laura Pollacci[♣], Emmanuele Chersoni[♣♠], Alessandro Lenci[♣]

[♣]CoLing Lab, Dipartimento di Filologia, Letteratura e Linguistica, University of Pisa (Italy)

[♠]Laboratoire Parole et Langage, Aix-Marseille University

{lucia.passaro|gianluca.lebani}@for.unipi.it, laurapollacci.pl@gmail.com, emmanuelechersoni@gmail.com, alessandro.lenci@ling.unipi.it

Abstract

English. This paper describes the CoLing Lab system for the EVALITA 2014 *SENTIment POLarity Classification* (SENTIPOLC) task. Our system is based on a SVM classifier trained on the rich set of lexical, global and twitter-specific features described in these pages. Overall, our system reached a 0.63 weighted F-score on the test set provided by the task organizers.

Italiano. *Questo contributo descrive il sistema CoLing Lab sviluppato per il task di SENTIment POLarity Classification (SENTIPOLC) organizzato nel contesto della campagna EVALITA 2014. Il nostro sistema è basato su un classificatore SVM addestrato sulle feature lessicali, globali e specifiche del canale twitter descritte in queste pagine. Il nostro sistema raggiunge uno score di circa 0.63 nel test set fornito dagli organizzatori del task.*

1 Introduction

Nowadays social media and microblogging services are extensively used for rather different purposes, from news reading to news spreading, from entertainment to marketing. As a consequence, the study of how sentiments and emotions are shown in such platforms, and the development of methods to automatically identify them, has emerged as a great area of interest in the Natural Language Processing community.

In this context, the research on sentiment analysis and detection of speaker-intended emotions from Twitter messages (tweets) appears to be a task on its own, rather distant from the previous sentiment classification research that focused on classifying longer pieces of texts, such as movie reviews (Pang and Lee, 2002).

As a medium, Twitter presents many linguistic and communicative peculiarities. A tweet, in fact, is a really short informal text (140 characters), in which the frequency of creative punctuation, emoticons, slang, specific terminology, abbreviations, links and hashtags is higher than in other domains. Twitter users post messages from many different media, including their cell phones, and they "tweet" about a great variety of topics, unlike what can be observed in other sites, which appear to be tailored to a specific group of topics (Go et al., 2009).

In this paper we describe the system we developed for the participation in the constrained run of the EVALITA 2014 *SENTIment POLarity Classification* Task (SENTIPOLC: Basile et al., 2014). The report is organized as follows: Section 2 describe the CoLing Lab system, starting from data preprocessing and annotation, to the adopted classification model. Section 3 shows the results obtained by our system.

2 System description

The CoLing Lab system for polarity classification of tweets includes the following three basic steps, that will be described in this section:

1. a **preprocessing** phase, aimed at the separate annotation of the linguistic and nonlinguistic elements in the target tweets;
2. a **feature extraction** phase, in which the relevant characteristics of the tweets are identified;
3. a **classification** phase, based on a Support Vector Machine (SVM) classifier with a linear kernel.

2.1 Data preprocessing and annotation

The aim of the preprocessing phase is the identification of the linguistic and nonlinguistic elements in the tweets and their annotation.

While the preprocessing of nonlinguistic elements such as links and emoticons is limited to their identification and classification (see section 2.2 for the complete list), the treatment of the linguistic material required the development of a dedicated rule-based procedure, whose output is a normalized text that is subsequently feed to a pipeline of general-purpose linguistic annotation tools. In details, the following rules applies in the linguistic preprocessing phase:

- Emphasis: tokens presenting repeated characters like *bastaaaa* are replaced by their most probable standardized form (i.e. *basta*).
- Links and emoticons: they are identified and removed.
- Punctuation: linguistically irrelevant punctuation marks are removed.
- Usernames: they are identified and normalized by removing the @ symbol and capitalizing the entity name.
- Hashtags: they are identified and normalized by simply removing the # symbol.

The output of this phase are "linguistically-standardized" tweets, that are subsequently POS tagged with the Part-Of-Speech tagger described in Dell'Orletta (2009) and dependency-parsed with the DeSR parser (Attardi et al., 2009).

2.2 Feature extraction

By exploiting the linguistic and non-linguistic annotations obtained in the preprocessing, a total of 1239 features have been extracted to be feed to the classifier. The inventory of features can be organized into the five classes described in this subsection.

2.2.1 Lexical features

Lexical features represent the occurrence of bad words or of words that are either highly emotional or highly polarized. Relevant lemmas were identified from two in-house built lexica (cf. below), and from Sentix (Basile and Nissim, 2013), a lexicon of sentiment-annotated Italian words.

ItEM. Lexicon of 347 highly emotional Italian words built by exploiting an online feature elicitation paradigm. Native speakers were requested to list nouns, adjectives or verbs that are strongly associated with the eight basic positive and negative emotions defined in Plutchik (2001): joy, trust, surprise, sadness, anger, disgust, fear and anticipation.

In our model, we used ItEM to compute, for each of the above mentioned emotions, the total count of strongly emotional tokens in each tweet.

Bad words lexicon. By exploiting an in house built lexicon of common Italian bad words, we reported, for each tweet, the frequency of bad words belonging to a selected list, as well as the total amount of these lemmas.

Sentix. Sentix (Sentiment Italian Lexicon: Basile and Nissim, 2013) is a lexicon for Sentiment Analysis in which 59,742 lemmas are annotated for their polarity and intensity, among other information. Polarity scores range from -1 (totally negative) to 1 (totally positive), while Intensity scores range from 0 (totally neutral) to 1 (totally polarized). Both these scores appear informative for our purposes, so that we derived, for each lemma, a Combined score C_{score} :

$$C_{score} = Intensity * Polarity$$

on the basis of which we organized the selected lemmas into the following five groups:

- strongly positives: $1 \leq C_{score} < 0.25$
- weakly positives: $0.25 \leq C_{score} < 0.125$
- neutrals: $0.125 \leq C_{score} \leq -0.125$
- weakly negatives: $-0.125 < C_{score} \leq -0.25$
- highly negatives: $-0.25 < C_{score} \leq -1$

Since Sentix relies on WordNet sense distinctions, it is not uncommon for a lemma to be associated with more than one $< Intensity, Polarity >$ pair, and consequently to more than one C_{score}. We decided to handle this phenomenon by identifying three different ambiguity classes and treating them differently. Lemmas with only one entry or whose entries are all associated with the same C_{score} value, are marked as "Unambiguous" and associated with that C_{score}. Ambiguous cases were treated by inspecting, for each lemma, the distribution of the associated C_{scores}.

Lemmas which had a Majority Vote[1] (MV) were marked as "Inferable" and associated with the C_{score} of the MV. If there was no MV, but the

[1] For each lemma a Majority Vote occurs when a class (strongly positive, weakly positive, etc) scores the greatest number of entries in Sentix. When two or more classes have the highest number of entries, the lemma has no MV.

highest number of senses in Sentix occurred simultaneously in both the positive or negative groups, lemmas were marked as "Inferable" and associated with the mean of the C_{scores}. All other cases were marked as "Ambiguous" and associated with the mean of the C_{scores}. To isolate a reliable set of polarized words, we focused only on the "Unambiguous" or "Inferable" lemmas and selected only the 250 topmost frequent according to the PAISÀ corpus (Lyding et al., 2014), a large collection of Italian web texts.

Other Sentix-based features in our model are: the number of tokens for each C_{score} group, the C_{score} of the first token in the tweet, the C_{score} of the last token in the tweet and the count of lemmas that are represented in Sentix.

2.2.2 Negation

Negation features have been developed to encode the presence of a negation and the morphosyntactic characteristics of its scope.

To count the negative tokens, we extracted from Renzi et al. (2001) an inventory of negative lemmas (e.g. "*non*") and patterns (e.g. "*non…mai*"), and counted the occurrence of these lemmas and structures in every tweet.

We then relied on the dependency parses produced by DeSR to characterize the scope of each negation, by assuming that the scope of a negative element is its syntactic head or the predicative complement of its head, in the case the latter is a copula.

Clearly, this has been a simplifying assumption, but in our preliminary experiments it shows to be a rather cost-effective strategy in the analysis of linguistically simple texts like tweets.

We included this information in our model by counting the number of negation pattern encountered in each tweet, where a negation pattern is composed by the PoS of the negated element plus the number of negative token depending from it and, in case it is covered by Sentix, either its Polarity, its Intensity and its C_{score} value. For instance, the negation pattern instantiated in the phrase *non tornerò mai* ("I will never come back") has been encoded, as "neg-negV_{POSPOL}", "neg-neg$V_{HIGHINT}$" and "neg-neg$V_{POSCOMB}$", meaning that a verb with high positive polarity, high intensity and a high C_{score} token is modified by two negative tokens.

2.2.3 Morphological features

The linguistic annotation produced in the preprocessing has been exploited also in the population of the following morphological statistics:

- number of sentences in the tweet;
- number of linguistic tokens;
- proportion of content words (nouns, adjectives, verbs and adverbs);
- number of tokens for Part-of-Speech.

2.2.4 Shallow features

This group of features has been developed to describe some distinctive characteristic of the web communication.

Emoticons. We built EmoLex, an inventory of common emoticons, such as :-(and :-), marked with their polarity score: 1 (positive), -1 (negative), 0 (neutral). In our system, EmoLex is used both to identify emoticons and to annotate their polarity.

In our model, emoticon-related features are the total amount of emoticons in the tweet, the polarity of each emoticon in sequential order and the polarity of each emoticon in reversed order. For instance, in the tweet *:-(quando ci vediamo? mi manchi anche tu! :*:** (":-(when are we going to meet up? I miss you, too :*:*") there are three emoticons, the first of which is negative while the others are positive. Accordingly, we feed our classifier with the information that the polarity of the first emoticon is -1, that of the second emoticon is 1 and the same goes for the third emoticon.

We additionally specified that the polarity of the last emoticon is 1, as it goes for that of the last but one emoticon, while the last but two has a polarity score of -1.

Links. We have performed a shallow classification of links using simple regular expressions applied to URLs. In particular, links are classified as following: video, images, social and other. For example, URLs containing substrings such as "youtube.com" or "twitcam" are classified as "video". Similarly URLs containing substrings such as "imageshack", or "jpeg" are classified as "images"., and URLs containing "plus.google" or "facebook.com" are classified as "social". Unknown links are inserted in the residual class "other".

We also use as feature the absolute number of links for each tweet.

Emphasis. The features report the number of emphasized tokens presenting repeated characters like *bastaaaa*, the average number of repeated characters in the tweet, and the cumulative number of repeated characters in the tweet.

For instance, in the message *Bastaaa! Sono stufaaaaaaaaa* ("Stop! I had enough"), there are 2 empathized tokens, the average number of repeated characters is 5, and the cumulative number of repetitions is 10.

Creative Punctuation. Sequences of contiguous punctuation characters, like "!!!", "!?!?!?!!?" or "……", are identified and classified as a sequence of dots, exclamations marks, question marks or mixed.

For each tweet, we mark the number of sequences belonging to each group and their average length in characters.

Quotes. The number of quotations in the tweet.

2.2.5 Twitter features

This group of features describes some Twitter-specific characteristics of the target tweets.

Topic. This information marks if a tweet has been retrieved via a specific political hashtag or keywords.

Usernames. The number of @username in the tweet.

Hashtags. We tried to infer the polarity of an hashtag by generalizing over the polarity of the tweets in the same thread. In other words, we used every hashtags we encountered as a search key [2] to download the most recent tweets in which they occur and inferred the polarity of the retrieved tweets by simply counting the number of positive and negative words in them.

In doing so, we made the assumption that the polarity of an hashtag is likely to be the same of the words it typically co-occurs with.

This, of course, does not take into account any kind of contextual variability of words meaning. We are aware that this is an oversimplifying assumption; nevertheless, we are confident that, in most cases, the polarity of the hashtag will reflect the polarity of its typical word contexts.

Moreover, tweets were assumed to be positive if they contained a majority of positive words, negative if they contained a majority of negative words, neutral otherwise.

In order to determine the polarity of a word, we used the scores of the Sentix lexicon. Words with a positive score ≤ 0.7got a score of 1, while words with a negative score ≤ -0.7received the score of -1. All the other words got a score of 0 (neutrality).

Unfortunately, for many hashtags in the corpus we have been able to retrieve just a small number of tweets, so that we chose to filter out those below a frequency threshold of 20 tweets, leaving us with 279 polarity-marked hashtags.

By relying on this hashtag-to-polarity mapping, the hashtag-related features in our model consisted in the total amount of hashtag for tweet, the polarity of each hashtag in sequential order and the polarity of each hashtag in reversed order.

2.3 Classification

Due to the better performance of SVM-based systems in analogue tasks (e.g. Nakov et al., 2013), we chose to base the CoLing Lab system for polarity classification on the SVM classifier with a linear kernel implementation available in Weka (Witten et al., 2011), trained with the Sequential Minimal Optimization (SMO) algorithm introduced by Platt (1998).

The classification task proposed by the organizers could be approached either by building two separate binary classifiers relying of two different models (one judging the positiveness of the tweet, the other judging its negativeness), or by developing a single multiclass classifier where the possible outcomes are Positive Polarity (Task POS:1, Task NEG:0), Negative Polarity (Task POS:0, Task NEG:1), Mixed Polarity (Task POS:1, Task NEG:1) and No Polarity (Task POS:0, Task NEG:0).

We tried both approaches in our development phase, and found no significant difference, so that we opted for the more economical setting, i.e. the multiclass one.

3 Experiments and Results

The evaluation metric used in the competition is the macro-averaged F_1-score calculated over the positive and negative categories. Our model obtained a macro-averaged F_1-score of 0.6312 on the test set and was ranked 3rd among 11 submissions. Table 2 reports the results of our model.

In addition, we present here two additional configurations (L and S) of our system, both of them using a smaller number of features.

The Lexical Model (L) is trained only on lexical features (see section 2.2.1), negation (see section2.2.2) and hashtags. This last group of features is used to train this model because the polarity of a thread is inferred from Sentix (see section 2.2.5).

The Shallow Model (S) is trained using only the non lexical features described in sections 0, 2.2.4, 2.2.5 (topic and usernames).

[2] We use the Python-Twitter library to query the Twitter API (https://code.google.com/p/python-twitter.)

Table 1 summarizes the features used to train the different models (F(ull), L(exical), S(hallow)), showing for each model the number of features:

Group	Features	#	F	L	S
Lexical	Badwords	28	∨	∨	
Lexical	ItEM	9	∨	∨	
Lexical	Sentix	1023	∨	∨	
Negation	Negation	53	∨	∨	
Morphol. features	Morphol. features	18	∨		∨
Shallow	Emoticons	17	∨		∨
Shallow	Emphasis	3	∨		∨
Shallow	Links	5	∨		∨
Shallow	Punctuation	6	∨		∨
Shallow	Quotes	1	∨		∨
Shallow	Slang	10	∨		∨
Twitter	Hashtags	63	∨	∨	
Twitter	Topic	1	∨		∨
Twitter	Usernames	2	∨		∨
Total number of features		**1239**	**1239**	**1176**	**63**

Table 1: Features used to train the models.

The Full model is trained on all the features described in the previous sections (1239 features).

Table 2 shows the detailed scores for each class both in the Positive and Negative tasks. It also points out the aggregate scores for each task and the overall scores.

Task	Class	Precision	Recall	F-score
POS	0	**0.7976**	0.7806	0.789
POS	1	0.581	**0.4109**	**0.4814**
POS task		0.6893	**0.5957**	**0.6352**
NEG	0	0.6923	0.6701	**0.681**
NEG	1	**0.6384**	0.5201	0.5732
NEG task		**0.6654**	0.5951	0.6271
GLOBAL		0.6774	**0.5954**	**0.6312**

Table 2: CoLing Lab system results

Table 3 shows the results obtained by the Lexical model, with 1176 features.

Task	Class	Precision	Recall	F-score
POS	0	0.7599	0.7755	0.7676
POS	1	0.4913	0.2981	0.371
POS task		0.6256	0.5368	0.5693
NEG	0	0.66	**0.6861**	0.6728
NEG	1	0.6218	0.4522	0.5237
NEG task		0.6409	0.5692	0.5983
GLOBAL		0.6333	0.553	0.5838

Table 3: CoLing Lab Lexical (L) system results

Table 4 reports the results obtained by the Shallow model, trained using non lexical information only, for a total of 63 features.

Task	Class	Precision	Recall	F-score
POS	0	0.7578	**0.8679**	**0.8092**
POS	1	**0.7184**	0.2205	0.3374
POS task		**0.7381**	0.5442	0.5733
NEG	0	**0.7369**	0.5174	0.608
NEG	1	0.5778	**0.6582**	**0.6154**
NEG task		0.6574	0.5878	0.6117
GLOBAL		**0.6978**	0.566	0.5925

Table 4: CoLing Lab Shallow (S) system results

4 Discussion

The best model to predict the polarity of a tweet is the one that combines lexical and shallow information (Full model).

Even though it achieves a better F_1-score, the global precision of the Shallow model is higher than the precision of the Full Model, despite the much smaller numbers of features. In particular, the Shallow model recognizes positive tweet more accurately. It is worth noticing that the class of positive tweets is the one in which our systems score worst. Besides the fact that the tweet class distribution is unbalanced in the training corpus, positive lexical features are likely to be not as able to predict tweets positivity, as negative features are with respect to negative tweets.

To sum up, on the one hand the three experiments demonstrate that significant improvements can be obtained by using lexical information. On the other hand the results highlight that the lexical coverage of the available resources such as Sentix and ItEM must be increased in order to obtain a more accurate classification.

5 Conclusion and future work

The CoLing Lab system participated in SENTIment POLarity Classification (SENTIPOLC) in EVALITA 2014 using a Support Vector Machine approach. The system combines lexical and shallow features achieving an overall F_1-score of 0.6312. Future developments of the system include refining the preprocessing phase, increasing the coverage of the lexical resources, improving the treatment of negation, and designing a more sophisticated way to exploit the information coming from the tweet thread. In particular, we are confident that a better preprocessed text and larger lexical resources will significantly enhance our system's performance.

Acknowledgments

Lucia C. Passaro received support from the Project SEMantic instruments for PubLIc admin-

istrators and CitizEns (SEMPLICE), funded by Regione Toscana (POR CReO 2007-2013), Gianluca E. Lebani works in the context of the PRIN grant 20105B3HE8, funded by the Italian Ministry of Education, University and Research; Emmanuele Chersoni is supported by the University Foundation A*MIDEX.

Reference

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi, and Joseph Turian (2009). Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In *Proceedings of EVALITA 2009*.

Valerio Basile and Malvina Nissim (2013). Sentiment analysis on Italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*: 100-107.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti and Paolo Rosso (2014). Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*.

Felice Dell'Orletta(2009). Ensemble system for Part-of-Speech tagging. In *Proceedings of EVALITA 2009*.

Alec Go, Richa Bhayani and Lei Huang (2009). *Twitter Sentiment Classification using Distant Supervision*. CS224N Project Report, Stanford.

Verena Lyding, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell'Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli (2014). The PAISÀ Corpus of Italian Web Texts. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*: 36-43.

Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov and Theresa Wilson (2013). Semeval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*: 79-86.

John C. Platt (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola (eds.) *Advances in Kernel Methods*: 185-208.

Robert Plutchik (2001). The Nature of Emotions. In *American Scientist*, 89: 344-350.

Lorenzo Renzi Gianpaolo Salvi and Anna Cardinaletti (2001). *Grande grammatica italiana di consultazione*. Il Mulino: Bologna.

Ian H. Witten, Elibe Frank, E and Mark A. Hall (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

10.12871/clicit2014216

The *FICLIT+CS@UniBO* System at the EVALITA 2014 Sentiment Polarity Classification Task.

Pierluigi Di Gennaro
DISI - University of Bologna, Italy
pierluigi.digennaro@gmail.com

Arianna Rossi
FICLIT - University of Bologna, Italy
ar.ariannarossi@gmail.com

Fabio Tamburini
FICLIT - University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

English. This paper presents a work in progress on the design of a sentiment polarity classification system that participates in the EVALITA 2014 SENTIPOLC task. Although we have been working on the system implementation for only three months, the results are promising, as the system ranked 5th (out of 9) in the subjectivity detection task and 7th (out of 11) in the sentiment polarity classification task.

Italiano. *Questo contributo presenta la progettazione di un sistema automatico per la classificazione della* sentiment polarity *che ha partecipato al task SENTIPOLC della campagna di valutazione EVALITA 2014. Nonostante i soli tre mesi di sviluppo, i risultati parziali sono promettenti in quanto il sistema si è classificato 5° (su 9) nel task di identificazione della soggettività e 7° (su 11) nel task relativo all'identificazione della polarità.*

1 Introduction

We developed two different approaches to Sentiment Polarity detection for the EVALITA 2014 SENTIPOLC task: (a) we started from the seminal paper (Basile, Nissim, 2013) and applied the same algorithm that had been proposed, but on a different lexicon, that was specifically developed for this system, and (b) we tried to devise more complex syntactically-driven polarity combination techniques.

In section 2 we describe the development of the annotated lexicon, in section 3 we illustrate the procedures applied by the proposed system, in section 4 we describe the system for the Subjectivity Classification task and, lastly, in section 5, we discuss the overall results obtained in the EVALITA 2014 Sentiment Polarity Classification task.

2 Sentiment-lexicon generation

Our lexicon was created by collecting words from various sources and was annotated using a semi-automatic polarity classification procedure. Sentiment polarity shifters were also taken into account and inserted into the lexicon.

2.1 Adjectives and Adverbs

We started by considering all the adjectives and adverbs extracted from the De Mauro - Paravia Italian dictionary (2000). All the glosses connected to the different senses of each lemma were automatically classified by using the online Sentiment Analysis API provided by *Ai Applied*[1]. This automatic procedure assigned either a positive or a negative polarity score to each lemma/sense pair in the intervals [-1,-0.5], for negative polarity, and [0.5,1], for positive polarity.

2.2 Nouns and Verbs

Although adjectives and adverbs are widely considered to be a primary source of subjective content in a text (Taboada *et al.*, 2011), also some nouns and verbs have a polarity value. We extracted nouns and verbs from Sentix (Basile, Nissim, 2013), since we expected those lemmas to be a selected choice of sentiment words, and used the automatic procedure seen above to classify their polarity.

2.3 Manual check

The polarity lexicon annotated with the automatic procedure described above was then inspected

[1] http://ai-applied.nl/sentiment-analysis-api

manually to clean it up. When the API had assigned a wrong polarity score, a value of 1.01 or -1.01 was assigned to the word, in order to clearly discriminate the automatic from the manually assigned values for future work. In addition, all the lemmas that had an objective value were left out and were not considered in our system, assigning to them a conventional polarity value equal to 0.

2.4 Everyday language and abbreviations

Lastly, the specific features of the informal language of social media were taken into account and all those words that our system could not identify from the tweets' development set were then extracted. By doing so, we were able to collect several words used in everyday language, i.e. *cazzata* (bullshit), *coglione* (moron), and abbreviations, i.e. *tt, nn* (not translatable), that were not yet included in our lexicon and assign a polarity value to them.

2.5 Sentiment polarity shifters

There are several linguistic phenomena that can cause a shift of the polarity of a word from one pole to the other or intensify its semantic intensity (Taboada *et al.*, 2011). Only negators and shifters were considered in the current approach, but others will be taken into account in our future research.

1. **Negators**: words like *non* (not), *nessuno* (nobody), *niente* (nothing), *nulla* (nothing), *mai* (never), etc. reverse the polarity of sentiment words (Polanyi, Zaenen, 2006). A value of -1 was assigned to negators, so that, in a sentence like *Non si vede bene* (You can not see well), *non* negates *bene* and flip its polarity from + 0,76 to -0,76.

2. **Intensifiers**: they increase or decrease the semantic intensity of the lexical item(s) they accompany (Taboada *et al.*, 2011). A positive percentage was assigned to amplifiers, whereas a negative one was assigned to downtoners, as shown in Table 1. This percentual value multiplies the polarity score of the opinion word, so if, for example, *felice* (happy) has a positive score of 0.84, *molto felice* (very happy) will have a positive score of: 0.84 × (1 + 0.25)= 1.05. The same procedure was applied to words accompanied by downtoners, so if, for instance, *grave* (serious) as a negative value of 0.7, *poco grave* (not very serious) will have a value of: -0.7 × (1 - 0.25)= -0.52.

Intensifiers	Value
completamente	+0.75
drasticamente	+0.50
molto	+0.25
abbastanza	-0.15
poco	-0.25
leggermente	-0.50

Table 1: Percentages for some positive and negative intensifiers

2.6 Context-dependent words

A large set of words do not have a positive or negative value *per se*, but, on the contrary, they can take a different value depending on the context they happen (Liu, 2012). For example, in an expression like *maniere forti* (strong-arm methods), *forte* (strong) has a negative meaning, whereas in *forte legame* (strong link) it has a positive one. Moreover, some of these words are objective in most domains, but they can acquire a subjective value in others. The word *poeta* (poet), for instance, can be objective, as in *Dante è stato un poeta del XIII secolo* (Dante was a poet of the 13th century), but can also have a subjective metaphorical meaning, as in *Luca scrive delle lettere bellissime. È proprio un poeta!* (Luca writes wonderful letters. He's really a poet!). We decided not to consider context-dependent words in our system since they need a more sophisticated approach that involves word sense disambiguation and metaphor detection.

3 System implementation

As a first step for the development of our sentiment polarity classification system, we implemented the algorithm proposed in the seminal paper (Basile, Nissim, 2013). Starting from their corpus of Italian tweets called TWITA, they developed a simple system which assigns one out of three possible values – positive, neutral or negative – to a given tweet. In order to assign the values, the system extracts the information from a polarity lexicon that was specifically developed thanks to various general lexical resources, namely SentiWordNet (Esuli, Sebastiani, 2006; Baccianella *et al.*, 2010), Multi-WordNet (Pianta *et al.*, 2002) and WordNet (Fellbaum, 1998). We developed the same algo-

rithm that was proposed in (Basile, Nissim, 2013), but we used instead the lexicon described in section 2, considering it as the starting point, or baseline, for any further improvement.
We can summarize the process in the following steps:

1. The system calculates the polarity score of each entry in the lexicon as the mean of the different word senses' scores.

2. Given a tweet, the system assigns a polarity score to each of its tokens by matching them to the lexicon.

3. The system calculates the polarity score of a complete tweet as the sum of the different polarity scores of its tokens: a polarity score greater than 0 indicates a positive tweet, a polarity score lower than 0 indicates a negative tweet, a polarity score equal to 0 indicates a neutral tweet.

In view of the results and thanks to the experience obtained from this development, we also tried to devise more complex syntactically-driven polarity combination techniques.

3.1 Token processing

Before proceeding with the syntactic analysis, we applied some rules of substitution or elimination to all those textual parts that were irrelevant to the classification task or that could hinder POS-tagging, lemmatization and parsing. In particular:

- a generic label "*URL*" replaced URLs (http://abc.org);
- character # and @ were removed from hashtags (#abc) and mentions (@abc);
- a generic label "*EMOPOS*" replaced positive emoticons (see table 2)
- a generic label "*EMONEG*" replaced negative emoticons (see table 2)

We added the labels "EMOPOS" and "EMONEG" to the lexicon, and associated them to a polarity score of 1.0 and -1.0 respectively.

3.2 Syntactic analysis

Our system relies on the TULE parser (Lesmo, 2007) to analyze the syntactic structure of a single tweet. TULE includes a tokenizer, a morphological analyzer, a PoS-tagger and a dependency parser. It takes a natural language sentence as input and returns a dependency tree that describes its syntactic structure. For each token identified, the parser output includes its PoS-tag, the lemma and other morphological information about it.

Label	Emoticon
EMOPOS	(: :) :] [: :-) (-: [-: :-] (; ;) ;] [; ;-) (-; [-; ;-] :-D :D :-p :p (=: ;=D :=) :S @-) XD
EMONEG	:( ): :-( )-: ;( ); :-[ ]-: ;-( )-; :[:( ):]: :[:| :/ |: /: :=(:=| :=[xo :| D: O:

Table 2: Emoticons' list.

As one would expect, we found some difficulties in using TULE on certain tweets, thus we added a few pre-processing and filtering steps:

- *special characters*: special characters (i.e. *$*) were replaced by their equivalent Italian word (i.e. *dollaro*).
- *shortened URLs*: due to limited tweet length, Twitter can cut an URL; these were removed from the tweets.

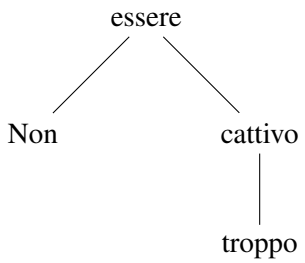
Our system uses adjacency lists (based on Boost library) with only one root node to represent dependency parser trees. Each node represents a token and contains all the relevant information about it: POS-tag, lemma, lexicon category (negator or intensifier) and polarity score. The system assigns a polarity score to a token by matching its lemma to the lexicon. If the lemma can not be found, three options are taken into account:

- *The polarity score of the lemma is 0*: a polarity score equal to 0 is conventionally assigned to the token.
- *The lemma is a polarity shifter*: the polarity score equals the intensification value of the shifter;
- *The lemma is not a polarity shifter*: the polarity score corresponds to the mean of the different word senses' scores.

When the polarity score of each tree node (i.e. each word in the sentence) has been calculated, the system assigns a polarity score to the whole tweet by applying a set of polarity propagation rules to the dependency tree. The system can choose between two options:

- *All tokens in a given sentence are not polarity shifters*: the polarity score is the sum of the polarity scores of each token.
- *One or more tokens in a given sentence are polarity shifters*: polarity shifters increase, decrease or reverse the polarity score of the item(s) linked to it. Starting from the polarity shifter that is closest to the leaves of the parse tree, the system sums the polarity score of the nodes linked to it and then multiplies this value by the polarity shifter's value.

For example the polarity score (PS) of the sentence *Non essere troppo cattivo* (Do not be too bad) is obtained as follows:



[(PS(*cattivo*) $\times$ (PS(*troppo*) + 1)) + PS(*essere*)] $\times$ PS(*Non*)

A tweet can be composed by more than one sentence. In this case, its final polarity score is obtained by summing all the polarity scores of its sentences.

Lastly, the system classifies a complete tweet as:

- *positive* if its polarity score is higher than 0;
- *neutral* if its polarity score is equal to 0;
- *negative* if its polarity score is lower than 0.

4 Subjectivity classification Task

Starting from the assumption that sentiment polarity and subjectivity classification are closely related, we used the results of our system described in section 3 to define whether a tweet is subjective or objective. Thus, we did not to implement a different system for subjectivity classification, but instead we derive subjectivity classification from sentiment polarity.

Given a tweet, it is classified as objective if its polarity score is equal to 0, otherwise it is classified as subjective. We are conscious that this is a coarse-grain approximation. If neutral tweets can only be objective, positive and negative tweets can be subjective or objective. We postponed the development of a better subjectivity classification system for further developments.

Rank	Combined F-score	F-score (0)	F-score (1)
1	0.7140	0.6005	0.8275
2	0.6871	0.5819	0.7923
3	0.6706	0.5344	0.8067
4	0.6497	0.4868	0.8127
-	<u>0.6134</u>	<u>0.4514</u>	<u>0.7755</u>
5	**0.5972**	**0.4480**	**0.7464**
6	0.5901	0.5031	0.6770
7	0.5825	0.4200	0.7451
8	0.5593	0.4424	0.6761
9	0.5224	0.3237	0.7211
10	*0.4005*	*0.0000*	*0.8010*

Table 3: Task 1 results – Constrained run, Subjectivity detection. In bold face the official results from the proposed system, underlined the results obtained using only the lexicon and in italics the baseline.

5 Results and discussion

Tables 3 and 4 present the results of the proposed system in the Subjectivity and Polarity Detection tasks respectively.

Although we have worked on the system implementation for only three months, the results are promising, as it ranked 5th (out of 9) in the subjectivity detection task and 7th (out of 11) in the sentiment polarity classification task. We did not participate in the irony detection task.

As we can see from Tables 3 and 4, our official results, produced by combining the new annotated lexicon with the complex algorithm for propagating lexical polarity values across dependency trees, do not exceed the unofficial results obtained by using only the lexicon.

The polarity propagation process is not problem-free and in the future we will consistently improve it, in order to obtain more reliable results. Also the lexicon must be improved: more lemmas must be inserted and the annotation schema can be enhanced by rethinking some of its features.

Rank	Combined F-score	Pos. Pol. F-score	Neg. Pol. F-score
1	0.6771	0.6752	0.6789
2	0.6347	0.6196	0.6498
3	0.6312	0.6352	0.6271
4	0.6299	0.6277	0.6321
-	<u>0.6062</u>	<u>0.5941</u>	<u>0.6184</u>
5	0.6049	0.6079	0.6019
6	0.6026	0.6153	0.5899
7	**0.5980**	**0.5940**	**0.6019**
8	0.5626	0.5556	0.5695
9	0.5342	0.5293	0.5390
10	0.5181	0.5021	0.5341
11	0.5086	0.5159	0.5013
12	*0.3718*	*0.3977*	*0.3459*

Table 4: Task 2 results – Constrained run, Polarity detection. In bold face the official results from the proposed system, underlined the results obtained using only the lexicon and in italics the baseline.

References

Baccianella S., Esuli A. and Sebastiani F. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of LREC 2010*, Valletta, Malta, 2200–2204.

Basile V. and Nissim M. 2013. Sentiment Analysis on Italian Tweets. *Proceedings of the 4th ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Atlanta, 100–107.

De Mauro T.. 2000. *Il dizionario della lingua italiana*. Paravia.

Esuli A. and Sebastiani F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. *Proceedings of LREC 2006*, Genova, 417–422.

Fellbaum C., editor. 2000. *WordNet. An Electronic Lexical Database*. The MIT Press.

Lesmo L. 1983. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47.

Liu B. 2012 Sentiment Analyses and Opinion Mining Synthesis Lectures on Human Language Technologies, 5,1 (2012): 1-167.

Pianta E., Bentivogli L. and Girardi C. 2002. MultiWordNet: developing an aligned multilingual database. *Proceedings of the First International Conference on Global WordNet*, Mysore, India, 21–25.

Polanyi L. and Zaenen A. 2006 Contextual Valence Shifters *Computing attitude and affect in text: Theory and applications*, Springer Netherlands, 1-10.

Taboada M., Brooke J., Tofiloski M., Voll K. and Stede M. 2011 Lexicon-Based Methods for Sentiment Analyses *Computational Linguistics*, 37(2):267–307.

Wiegand, M., Balahur, A., Roth, B., Klakow, D. and Montoyo, A. 2010 A survey on the role of negation in sentiment analysis *Proceedings of the ACL workshop on negation and speculation in natural language processing*, 60-68.

10.12871/clicit2014217

A Multiple Kernel Approach for Twitter Sentiment Analysis in Italian

Giuseppe Castellucci(†), **Danilo Croce**(‡), **Diego De Cao**(‡) and **Roberto Basili**(‡)
(†) Dept. of Electronic Engineering,
(‡) Dept. of Enterprise Engineering,
University of Roma, Tor Vergata
Via del Politecnico 1, Rome, 00133, Italy
{castellucci}@ing.uniroma2.it, {croce,decao,basili}@info.uniroma2.it

Abstract

English. This paper describes the UNITOR system that participated to the *SENTIment POLarity Classification* task within the context of Evalita 2014. The system has been developed as a workflow of Support Vector Machine classifiers. Specific features and kernel functions have been used to tackle the different sub-tasks, i.e. Subjectivity Classification, Polarity Classification and the pilot task Irony Detection. The system won 3 of the 6 evaluations carried out by the task organizers, and in the worst case it ranked in 4^{th} position w.r.t. about 10 participants.

Italiano. *Questo articolo descrive il sistema UNITOR che è stato valutato nel task di SENTIment POLarity Classification ad Evalita 2014. Il riconoscimento del sentimento nei Tweet è basato su un workflow di classificatori di tipo Support Vector Machine (SVM), il cui flusso è stato studiato appositamente per risolvere i diversi task proposti nella competizione. Rappresentazioni vettoriali specifiche sono state definite per modellare i tweet al fine di applicare funzioni Kernel che vengono utilizzate dai classificatori SVM. Il sistema ha ottenuto risultati promettenti risultando vincitore di 3 dei 6 task proposti.*

1 Introduction

Modern Internet technologies allow users to generate new contents, writing their opinions about facts, things and events. The interest in the analysis of the user-generated contents is rapidly growing. In particular, Sentiment Analysis (SA) of web data produced by users is becoming a crucial component for companies or politicians in order to check the mood on the web, and consequently adjust their strategies. Twitter[1] is one of the most popular social networking service that allows people to express themselves with very short messages. SA in Twitter represents a challenging task, as messages are short, informal and characterized by their own particular language, e.g. retweets ("*RT*"), user references ("@"), hashtags ("*#*") or other typical web slang, e.g. emoticons. Classical approaches to Sentiment Analysis (Pang et al., 2002; Pang and Lee, 2008) mainly focus on longer texts, e.g. movie reviews, resulting in performance drops when applied on tweets. Examples of tweet modeling within Machine Learning settings for the Twitter SA can be found in (Pak and Paroubek, 2010; Zanzotto et al., 2011; Kouloumpis et al., 2011; Agarwal et al., 2011; Croce and Basili, 2012; Castellucci et al., 2013; Rosenthal et al., 2014).

In this paper, the UNITOR system participating in the *Sentiment Polarity Classification* (SENTIPOLC) task (Basile et al., 2014) within the Evalita 2014 evaluation campaign is described. The system faces three proposed subtasks: *Subjectivity Classification*, *Polarity Classification* and the pilot task called *Irony Detection*. As the specific labeling of the challenge is rich and complex, we decomposed the analysis in different stages. The labeling of each tweet is determined by the application of a workflow of Support Vector Machine (Vapnik, 1998) classifiers. In this work, several kernel functions have been exploited to tackle the different nature of each subtask. The UNITOR system ranked among the 1^{st} and 4^{th} position in all the submitted runs, resulting the winning system in 3 of 6 evaluations.

In the rest of the paper, in Section 2 the classifiers, in terms of features, kernels are described and the adopted workflow is presented. In Section 3 the performance measures of the system are reported while Section 4 derives the conclusions.

[1] http://www.twitter.com

2 System Description

The UNITOR system participated to all the subtasks proposed in the SENTIPOLC (Basile et al., 2014) challenge: *Subjectivity Classification*, *Polarity Classification* and the pilot task *Irony Detection*. The first task aims at evaluating the performance of systems in capturing whether a message conveys a subjective position. The second task is intended to verify if a system is able to detect the polarity of a message, in terms of positive, neutral or negative classes. The last one is intended to verify the presence of irony.

2.1 Feature engineering

In our Supervised Learning setting, a multiple-kernel based approach has been adopted to acquire the SVM classifiers (Shawe-Taylor and Cristianini, 2004): the similarity between training and testing example is measured by kernel functions, that are applied to different feature representations, each engineered to capture different properties of each message.

First, all tweets have been processed through an adapted version of a Chaos natural language parser (Basili and Zanzotto, 2002). A normalization step is exploited before applying the Natural Language Processing chain. The following set of actions is performed: fully capitalized words are converted in their lowercase counterparts; hyperlinks are replaced by the token `LINK`; any character repeated more than three times are cleaned, as they cause high levels of lexical data sparseness (e.g. "*nooo!!!!!*" is converted into "*noo!!*"); all emoticons are replaced by special tokens[2].

Then, a set of feature vector is generated to let the SVM classifiers capture semantic properties of each tweet. In the rest of this Section, the representations of tweets are described.

Bag-Of-Word (`BOW`) is a representation that aims at capturing the lexical overlap between examples. A feature vector in which each dimension represents a lemma and a part-of-speech is derived from a tweet message. A boolean weighting is applied, i.e. a feature has a 1.0 value if the corresponding lemma and part-of-speech pair appears in the message.

SentixSum (`SSUM`) is a feature vector that is obtained using the Sentix (Basile and Nissim, 2013) lexicon. It is obtained aligning different existing resources. It consists of about 60.000 entries, each characterized by an Italian lemma, part-of-speech, WordNet (Miller, 1995) synset ID, and different polarity scores. Given a tweet, we derived the `SSUM` vector, as a 4-dimensional vector where each feature corresponds to the sum, with respect to each word, of the polarity scores that are available in the Sentix lexicon: positivity, negativity, polarity and intensity scores. The final vector is then normalized.

SentixDifference (`SDIFF`) is a feature vector describing how discordant are the words in a message. Again, this vector is obtained using the Sentix resource (Basile and Nissim, 2013). The `SDIFF` vector is 4-dimensional, and it reflects the 4 scores that can be extracted from this lexicon. In particular, each dimension is the result from the difference computed between the vectors of the maximally polar word and the minimally polar word. Formally, given $\vec{w_1}$ and $\vec{w_2}$ as the vectors in Sentix, representing the words respectively with the *maximum* and *minimum* polarity score respectively, then the `SDIFF` vector is computed as $sd(\vec{w_1}, \vec{w_2}) = \vec{w_1} - \vec{w_2}$.

Latent Semantic Analysis (`LSA`) representation aims at generalizing lexical information available through the `BOW` model. A vector representation for words is obtained from a co-occurrence Word Space built accordingly to the methodology described in (Sahlgren, 2006). A word-by-context matrix M is obtained through the analysis of a large scale corpus of 3 million of tweets. Each dimension is weighted through the Pointwise Mutual Information between a word and its context in a window of 3 words before or after. The *Latent Semantic Analysis* (Landauer and Dumais, 1997) technique is then applied as follows. The matrix M is decomposed through Singular Value Decomposition (SVD) (Golub and Kahan, 1965) into the product of three new matrices: U, S, and V so that S is diagonal and $M = USV^T$. M is then approximated by $M_k = U_k S_k V_k^T$, where only the first k columns of U and V are used, corresponding to the first k greatest singular values. The original statistical information about M is captured by the new k-dimensional space, which preserves the global structure while removing low-variant dimensions. Every word of a tweet is projected in the reduced Word Space and a message is represented by applying an *additive linear combination*. Only verbs, adjectives, nouns and hashtags are considered.

[2] We normalized 113 well-known emoticons in 13 classes.

IronyVector (IV) is a specific vector designed to capture the irony of messages. It has been inspired by some recent works on irony detection (Carvalho et al., 2009; Reyes et al., 2012). This is a 7-dimensional vector in which each value aims at capturing some linguistic feature of ironic messages. The features are the following: *hasQuotationMarks*, if the tweet contains a quotation mark; *hasQuestionMarks* if the message contains a question mark; *hasExclamationMarks* if the tweet contains an exclamation mark; *lastTokenIsAPunctuation* if the last token of a message is a punctuation; *lastTokenIsAHappySmile* if a tweet ends with a smile belonging to the *happy* category with respect to our classification; *lastTokenIsASadSmile* if last token is a sad smile; *lastTokenIsASmile* if message ends with a smile. Each activated dimension is boolean weighted, i.e. the value is 1.0.

Out-of-Topic Weighted BOW (WBOW) is a Bag-Of-Word vector representing the words in a message. The main difference with respect to the previous BOW representation is the adopted weighting scheme. In fact, in this case we leverage on the Word Space previously described. For each dimension representing a lemma/part-of-speech pair, its weight is computed as the cosine similarity between the LSA vector of the considered word and the vector obtained from the linear combination of all the other words in the message. This vector aims at capturing how a word is out of context in a sentence, and therefore it should help in capturing unconventional use of words, and it should be an indicator of an ironic use of language.

LSAIrony (LSAIR) is a 4-dimensional vector specifically designed for the irony detection tasks. Its purpose is to compute a measure of dissimilarity between the words in a tweet, exploiting, again, the idea that an ironic message makes an unconventional use of words. Each dimension is a measure of how much words are dissimilar in a specific grammatical category. Thus, the first dimension measures the dissimilarity in the Word Space of the verbs, the second dimension considers nouns, the third look at the dissimilarities between adjectives, while the last dimension takes into account all the words of the message.

2.2 A Cascade of SVM classifiers for Sentiment Analysis

In Figure 1 the workflow of SVM classifiers developed for the SENTIPOLC task is shown. Each tweet is pre-processed and feature vectors are generated as described in the previous Section. Separated representations are considered in the *constrained* and *unconstrained* settings. In the constrained setting only feature vectors using tweet information or public available lexica are considered. In the unconstrained setting, feature vectors are derived also by exploiting other tweet messages, that are used in the acquisition of the Word Space (LSA and LSAIR).

Each tweet, in terms of its multi-vector representation, is then fed to the classifiers, and it flows over the cascade following the diagram in Figure 1. At the end of the workflow, 7 possible outputs are allowed according to the specification of the task. A binary code is used to express the different outputs: 4 bits are used to express the *subjectivity*, *positivity*, *negativity* and *irony* of a message. For example, a tweet that is subjective, and expresses both a positive and negative sentiment is labeled as 1110.

In the following, the specific kernel functions used in each classification stage are reported.

Subjective classifier. At the first stage of the workflow, the *Subjectivity* classifier is invoked. This is a crucial step, as an error in the classification of the subjectivity of the message compromises the entire cascade. At this stage, the linear combination of a linear kernel is applied over the BOW and the SSUM vectors. In the unconstrained case, a 2-degree polynomial kernel (Shawe-Taylor and Cristianini, 2004) is applied on the BOW representation in combination with a linear kernel on SSUM and a linear kernel on LSA.

Explicit polarity classifier. Here, the classifier adopts the same representations and kernels that have been used for the Subjective classifier. Consequently, the resulting classification function only depends on the labels of the training material.

Explicit positive/negative classifier. Again, the same setting used in the previous classifiers is exploited. Instead of a single binary classifier discriminating between two classes (i.e. positive and negative), here we have two binary classifiers. This is necessary to enable the labeling of tweets conveying both a positive and negative polarity in opposition of a neutral polarity. This last labeling is assigned when both the explicit positive and negative classifiers express a negative confidence of the classification.

Irony classifier. When a tweet does not explic-

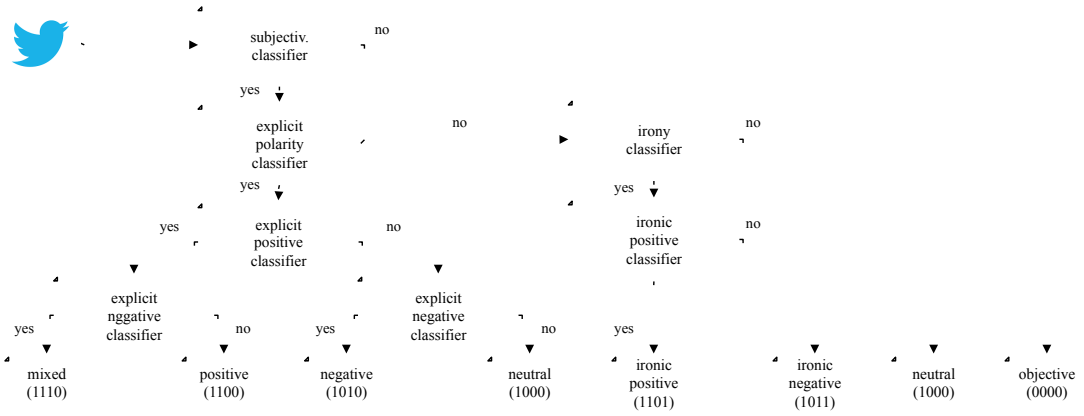


Figure 1: The UNITOR classifier workflow

itly express a sentiment, it may be ironic. It is reflected in the workflow as a classifier that separated ironic and neutral tweets. In the constrained case, the irony classifier adopts a BOW vector representation with a linear kernel combined with the SDIFF representation, again with a linear kernel. In the unconstrained case, a linear kernel applied on the WBOW representation is combined with a 2-degree polynomial kernel on the BOW vector and a linear kernel on the SDIFF vector.

Ironic positive/negative classifier. When a tweet is ironic, the last classification stage adopts more representations both in the constrained and in the unconstrained case. In the former, a linear kernel is applied on the BOW, SDIFF and IV vector. In the unconstrained case, the representations involved are: BOW, SDIFF, IV, LSAIR with a linear kernel, and the LSA with a RBF kernel (Shawe-Taylor and Cristianini, 2004). When training the explicit positive/negative and ironic positive/negative classifiers, the training material was split according the presence of irony as it affects also the way of expressing the polarity.

Each classifier is built by using a custom Java Support Vector Machine (SVM) implementation based on LibSVM (Chang and Lin, 2011). This implementation is specifically developed to support the combination of multiple representations and kernels. The Figure 1 reflects also the learning strategy that has been set up during the tuning phase: each classifier has been trained on the specific subset of the data of interest. Parameter tuning phase has been done by a fixed 80/20 split of the training data. Training data have been downloaded through the web interface proposed by the organizers[3], resulting in 4,033 tweet that were available at the time of the download. We lost 482 messages during the download phase due to Twitter policies. More information about the data, annotation process and evaluation metrics can be found in (Basile et al., 2014).

3 Results

In this Section the results of the UNITOR system are reported. Performance measures refer to the three subtasks proposed in the SENTIPOLC evaluation. Test data were downloaded through the same web interface provided by the organizers. Even for test data, some messages were no more available due to Twitter policies. Test data were supposed to be 1,938, while we downloaded 1,752 tweets. In Table 1 cumulative F1 scores and ranks for the UNITOR system are reported. Detailed performances are reported in the rest of the Section.

	C	U
Subjectivity Classification	68.7 (2)	69.0 (1)
Polarity Classification	63.0 (4)	65.5 (2)
Irony Detection	57.6 (1)	59.6 (1)

Table 1: UNITOR overall score and ranks. C and U refer to constrained and unconstrained runs

In Tables 2 and 3 the performances of the *Subjectivity Classification* subtask are reported. Both the constrained and unconstrained runs are here presented. UNITOR performances are remarkable as in the constrained run it ranks in 2^{nd} position, while in the unconstrained one is in 1^{st} position. In the constrained case, representations adopted are able to correctly determine whether a message is subjective with good precision, as demonstrated by the *Subjective* precision measure.

[3] http://www.di.unito.it/~tutreeb/sentipolc-evalita14/tweet.html

However, the winning system here was about 3 points ahead, in particular resulting more effective in the detection of non-subjective messages. The UNITOR system is not able to tackle messages that are too short. For example, some tweets were composed only by one or two words. In such messages there is not enough information for our classifiers. In the unconstrained case, the contribution of the LSA vector representation is demonstrated by the higher score obtained with respect to the constrained case. This makes the UNITOR system one of the best performing system in detecting the subjectivity of messages.

NotSubjective			Subjective		
P	R	F1	P	R	F1
57.7	**58.7**	58.2	**85.8**	73.6	79.2

Table 2: Subjectivity classification: constrained

NotSubjective			Subjective		
P	R	F1	P	R	F1
60.6	54.9	**57.6**	84.9	76.2	80.3

Table 3: Subjectivity classification: unconstrained

In Tables 4 and 5 the performances for the *Polarity Classification* are reported. In the constrained case, the results are comparable with the best systems, i.e. less than 5 points from the 1^{st} system. Analyzing the full results, our main problems are in the detection of the positive polarity classes, as we observed a 15 point drop of precision in the positive class. In the unconstrained case, the contribution of our tweet-specific Word Space derived vectors is again remarkable. In this case the UNITOR system is able to have the best performances in all the measures for the positive class (except the recall for the positive class). In the case of the negative class the system is not able to perform as well as the positive case. However, we consider this result very promising as the improvement w.r.t. our constrained run is about of 3 points. It means that the unsupervised analysis of a large tweet corpus is beneficial even for the polarity classification task. In this task, many misclassifications affect messages characterized by an implicit inversion of polarity. Moreover, messages that were not correctly recognized as ironic by the Explicit polarity classifier determine a more complex classification in the *Polarity Classification* stage, as we have a separated classifier for polarity in the ironic case.

In Tables 6 and 7 the performances of the UNITOR system on the pilot task *Irony Detection*

Positivity						
P_0	R_0	$F1_0$	P_1	R_1	$F1_1$	F1
79.5	77.0	78.2	56.0	40.9	47.3	62.8
Negativity						
P_0	R_0	$F1_0$	P_1	R_0	$F1_1$	F1
72.2	60.1	65.6	61.4	60.2	60.8	63.2

Table 4: Polarity classification: constrained

Positivity						
P_0	R_0	$F1_0$	P_1	R_1	$F1_1$	F1
82.1	**77.5**	**79.7**	**60.8**	48.2	**53.7**	**66.7**
Negativity						
P_0	R_0	$F1_0$	P_1	R_0	$F1_1$	F1
73.8	59.9	66.2	62.1	**62.4**	62.2	64.2

Table 5: Polarity classification: unconstrained

are reported. In the constrained case, the UNITOR system reaches the 1^{st} position on the rank with a combined F1 score of 57.59. The system performs very well in detecting not-ironic messages, as demonstrated by the *NotIronic* columns. Probably this is due to the unbalanced dataset provided for this task. In fact, only 564 over 4515 messages in the training data were labelled as ironic. If the same ratio was in the test set, it can be seen as a bias for the evaluation. In the unconstrained case, the UNITOR system reaches again the 1^{st} position in the rank. The contribution of the unconstrained representations helped, as a gain of 2 points in the combined F1 score has been observed. Moreover, representations used in the unconstrained case allow to be more precise when a message is ironic, as the 4 points precision increment suggests. However, a drop in recall makes the two systems perform more or less the same in terms of Ironic F1 measure (about 35 points in F1 score in both cases).

NotIronic			Ironic		
P	R	F1	P	R	F1
93.1	69.6	79.6	26.6	**52.9**	**35.5**

Table 6: Irony detection: constrained

NotSubjective			Subjective		
P	R	F1	P	R	F1
92.1	**76.3**	**83.5**	**30.6**	**42.9**	**35.7**

Table 7: Irony detection: unconstrained

4 Conclusions

In this paper the description of the UNITOR system participating to the SENTIPOLC task at Evalita 2014 has been provided. The system won 3 of the 6 evaluations carried out in the task, and in

the worst case it ranked in the 4^{th} position. Thus, the proposed classification strategy is one of the best performing in the Twitter Italian Sentiment Analysis scenario. The UNITOR system won the Irony Detection task both in constrained and unconstrained settings. Even if the evaluation dataset for this subtask was quite small, the irony specific features that were studied for this problem were able to detect irony in short messages. However, further work is needed to improve the overall (low) F1 scores. The nature of Twitter messages does not help, as tweets are very short and the amount of useful information for detecting irony is often out of the message. For these reasons, we think that more information can be extracted using message contexts, as demonstrated in (Vanzo et al., 2014b; Vanzo et al., 2014a) for the English and Italian languages.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Ws on Languages in Social Media*, pages 30–38. ACL.

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proceedings of the 4th Ws: Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation campaign of NLP and Speech tools for Italian (EVALITA)*, Pisa, Italy.

Roberto Basili and Fabio Massimo Zanzotto. 2002. Parsing engineering and empirical robustness. *Nat. Lang. Eng.*, 8(3):97–120.

Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: Oh...!! it's "so easy" ;-). In *1st CIKM WS on Topic-sentiment Analysis for Mass Opinion*, pages 53–56. ACM.

Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2013. Unitor: Combining syntactic and semantic kernels for twitter sentiment analysis. In *2nd Joint Conf. *SEM: Vol. 2: Proceedings of SemEval*, pages 369–374. ACL.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Danilo Croce and Roberto Basili. 2012. Grammatical feature engineering for fine-grained ir tasks. In *IIR*, pages 133–143.

Gene Golub and W. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Numerical Analysis*, 2(2):pp. 205–224.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In *ICWSM*.

Tom Landauer and Sue Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of LREC*. ELRA.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP, vol. 10*, pages 79–86. ACL.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data and Knowledge Engineering*, 74(0):1 – 12.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th SemEval WS*, pages 73–80. ACL and Dublin City University.

Magnus Sahlgren. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Andrea Vanzo, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2014a. A context based model for twitter sentiment analysis in italian. In *Proceedings of CLIC (To Appear)*, Pisa, Italy, December.

Andrea Vanzo, Danilo Croce, and Roberto Basili. 2014b. A context-based model for sentiment analysis in twitter. In *Proceedings of COLING*, pages 2345–2354. ACL and Dublin City University.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Kostas Tsioutsiouliklis. 2011. Linguistic redundancy in twitter. In *EMNLP*, pages 659–669.

10.12871/clicit2014218

Relying on intrinsic word features to characterise subjectivity, polarity and irony of Tweets*

Francesco Barbieri, Francesco Ronzano, Horacio Saggion
Universitat Pompeu Fabra, Barcelona, Spain
name.surname@upf.edu

Abstract

English. We describe our participation to the SENTIPOLC task of EVALITA 2014. We experimented the use of intrinsic word features to characterise each Tweet. We relied only on these features to train a set of Decision Trees to characterise the subjectivity, the polarity and the ironic contents of each Tweet. In Task 1 and Task 2 our model shows good performances comparing to the other participants, even if there is still space for improvements. In Task 3 our model do not achieve acceptable performances. We interpret and discuss these results.

Italiano. *Descriviamo la nostra partecipazione a SENTIPOLC di EVALITA 2014. Abbiamo sperimentato l'uso di features intrinseche delle parole per caratterizzare ogni Tweet. Grazie a queste features abbiamo costruito Decision Trees per determinare la soggettività, la polarità e il contenuto ironico di ogni Tweet. Nel Task 1 e Task 2 il nostro modello mostra buone prestazioni rispetto agli altri partecipanti, anche se c'è ancora spazio per miglioramenti. Nel Task 3 il nostro modello non raggiungere prestazioni accettabili. Nel paper discutiamo tali risultati fornendo possibili interpretazioni.*

1 Motivation

The automatic identification of the diverse facets of sentiments and opinions expressed by social media users constitutes a relevant and challenging research trend. In this context, the Sentiment Polarity Classification task of EVALITA 2014 (SENTIPOLC, Basile et al. (2014)) offers both a shared dataset and a venue to experiment and compare new approaches to the analysis of opinionated texts in social media. SENTIPOLC proposes three tasks respectively devoted to automatically determine the subjectivity, the polarity and the irony of a Tweet. This paper describes our participation in these three SENTIPOLC taks. We exploited an extended version of the Tweet classification features and approach described in Barbieri et al. (2014). In particular, we experimented the use of intrinsic word features, characterising each word in a Tweet (like usage frequency in a reference corpus, number of associated synsets, etc.), to try to model and thus automatically determine its subjectivity, polarity and ironic traits. We did not exploit textual features (like word occurrences, bigrams, skipgrams or other word patterns) to try to reduce the dependency of our model on a specific topic or on the set of words used in the considered domain. We aim to detect two aspects of Tweets by intrinsic word features: the style used (e.g. register used, frequent or rare words, positive or negative words, etc.) and the unexpectedness in the use of words, particularly important for subjectivity and irony (Lucariello, 1994). We exploited Decision Trees to classify Tweets in all the three SENTIPOLC tasks. In Section 2 we describe the tools we used to process Tweet contents. In Section 3 we introduce the features we built our model on. Section 4 analyses the performances of our model concerning the three tasks of SENTIPOLC.

2 Text Analysis and Tools

In order to process the text of Tweets so as to enable the feature extraction process, we used a set of freely available tools. First of all, we associated to each Tweet a normalised version of its text by expanding abbreviations and slang expressions, deleting emoticons, properly converting hashtags

*The research described in this paper is partially funded by the Spanish fellowship RYC-2009-04291, the SKATER-TALN_UPF project (TIN2012-38584-C06-03), and the EU project Dr. Inventor (n. 611383).

into words whether they have a syntactic role. We then tokenised, PartOfSpeech-tagged, applied Word Sense Disambiguation (UKB) and removed stop words from the normalized text of Tweets by exploiting Freeling (Carreras et al., 2004). We also used the Italian WordNet 1.6[1] to get synsets and synonyms of each word of a Tweet as well as the sentiment lexicon Sentix[2] (Basile and Nissim, 2013) derived from SentiWordnet (Esuli and Sebastiani, 2006) to get the polarity of synsets. We relied on the CoLFIS Corpus of Written Italian[3] to obtain the usage frequency of words in written Italian. We exploited the results of these analyses of the contents of Tweets to generate the word intrinsic features we describe in Section 3.

3 Our Model

In the three tasks of SENTIPOLC, we trained a Decision Tree to classify Tweets as far as concern their subjectivity, polarity and ironic contents. We exploited the widespread machine learning framework Weka in order to train and test our classification models. We characterised each Tweet by six classes of features all describing intrinsic aspects of the words of the same Tweet. These feature classes are: Frequency, Synonyms, Ambiguity, Part of Speech, Sentiments, and Punctuation.

3.1 Frequency

We accessed the CoLFIS Corpus to retrieve the frequency of each word of a Tweet. Thus, we derive three types of Frequency features: *rarest word frequency* (frequency of the most rare word included in the Tweet), *frequency mean* (the arithmetic average of all the frequency of the words in the Tweet) and *frequency gap* (the difference between the two previous features). These features are computed including all the words of each Tweet. We also determined these features by considering only Nouns, Verbs, Adjectives, and Adverbs.

3.2 Synonyms

We consider the frequencies (in CoLFIS Corpus) of the synonyms of each word in the Tweet, as retrieved from the Italian WordNet 1.6. Then we computed, across all the words of the Tweet: the *greatest / lowest number of synonyms* with frequency higher than the one present in the Tweet, the *mean number of synonyms* with frequency greater / lower than the frequency of the related word present in the Tweet. We determine also the greatest / lowest number of synonyms and the mean number of synonyms of the words with frequency greater / lower than the one present in the the Tweet (*gap* feature). We computed the set of Synonyms features by considering both all words of the Tweet together and only words belonging to each one of the four Parts of Speech listed before.

3.3 Ambiguity

To model the ambiguity of the words in the Tweets we use the WordNet synsets associated to each word. Our hypothesis is that if a word has many meanings (synset associated) it is more likely to be used in an ambiguous way. For each Tweet we calculate the *maximum number of synsets* associated to a single word, the *mean synset number* of all the words, and the *synset gap* that is the difference between the two previous features. We determine the value of these features by including all the words of a Tweet as well as by considering only Nouns, Verbs, Adjectives or Adverbs.

3.4 Part Of Speech

The features included in the Part Of Speech (POS) group are designed to capture the style of the Tweets. The features of this group are eight and each one of them counts the number of occurrences of words characterised by a certain POS. The eight POS considered are *Verbs*, *Nouns*, *Adjectives*, *Adverbs*, *Interjections*, *Determiners*, *Pronouns*, and *Appositions*.

3.5 Sentiments

The sentiments of the words in Tweets are important for two reasons: to detect the *sentiment* style (e.g. if Tweets contain mainly positive or negative terms) and to capture unexpectedness created by a negative word in a positive context and viceversa. Relying on Sentix (see Section 2) we computed the *number of positive / negative words*, the *sum of the intensities of the positive / negative scores of words*, the *mean of positive / negative score of words*, the *greatest positive / negative score*, the *gap between the greatest positive / negative score and the positive / negative mean*. As previously done, we computed these features by considering only Nouns, Verbs, Adjetives, and Adverbs.

[1] http://multiwordnet.fbk.eu/english/home.php
[2] http://www.let.rug.nl/basile/twita/sentix.php
[3] http://linguistica.sns.it/CoLFIS/Home_eng.htm

3.6 Punctuation

We also want to capture the punctuation style of the authors of a Tweet. Punctuation is very important in social networks: a full stop at the end of a subjective message may change the polarity of the message, the use of ellipses can be sign of irony (Carvalho et al., 2009). Each feature that is part of the Punctuation set is the number of a specific punctuation mark, including: ".", "#", "!", "?", "$", "%", "&", "+", "-", "=", "/".

	P	R	F1
Task 1 (subj.)	0.7332	0.6011	0.6497
Task 2 (polarity)	0.6565	0.5723	0.6049
Task 3 (irony)	0.5797	0.4591	0.4987

Table 1: Final scores (arithmetic average of the score of each class) of the three tasks organised in Precision, Recall and F-Measure.

4 Experiments and Results

In this section we present our results in the three SENTIPOLC tasks (see Table 1). We only report final results (mean of Precision, Recall and F-Measure of each class). In order to get other participants results, please refer to the SENTIPOLC paper (Basile et al., 2014).

4.1 Task 1: Subjectivity Classification

Given a message, decide whether the message is subjective or objective. Our model scores at position four out of nine groups. Our score is six points less than the best one in F-measure. Our system showed that we can determine if a Tweet is subjective or not with an acceptable precision by not considering explicitly words or word patterns, but only relying on intrinsic word features.

4.2 Task 2: Polarity Classification

Given a message, decide whether the message is of positive, negative, neutral or mixed sentiment (i.e. conveying both a positive and negative sentiment). In this task our model ranks fifth out of eleven participants. We obtain an averaged F-measure of 0.6049.

4.3 Task 3: Irony Detection

Given a message, decide whether the message is ironic or not. At this task our system scored as the last one, clearly showing that, at least for the Tweet dataset exploited in SENTIPOLC, relying only on intrinsic word features has limited power in determining if a Tweet is ironic or not.

5 Conclusions

In this paper we describe our participation to the SENTIPOLC task of EVALITA 2014. We experimented the use of intrinsic word features to characterise each Tweet. We relied exclusively on these features to train a set of Decision Trees respectively useful to determine the subjectivity, polarity and irony in Tweets. We explicitly decided not to rely on explicit words or word patterns as features. In Task 1 and Task 2 our model shows good performances comparing to other models, even if there is still space for improvements. In Task 3 our model do not achieve acceptable performances. Among other considerations, we related this issue to the fact that the training data in SENTIPOLC are strongly dependent on a specific topic, politics and this topic dependence limits the effectiveness of our system. In fact our classifier does not use words or word patterns that usually constitute features exploited to characterise a domain. In general, we noticed that avoiding text features may constitute a limitation for a classifier if the dataset to deal with concerns a specific topic and thus topic specific words could constitute good features to model the domain. As future work we are planning to experiment with other classification approaches (Support Vector Machines among them) as well as to evaluate the utility to complement the feature set we presented in this paper with word and word pattern features (like word occurrences, bigrams, skip-grams or other word patterns).

References

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2014. Italian Irony Detection in Twitter, a First Approach. In *Proceedings of the First Italian Conference of Computational Linguistic*, Pisa, Italy, December.

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation cam-*

paign of Natural Language Processing and Speech tools for Italian (EVALITA'14), Pisa, Italy.

Xavier Carreras, Isaac Chao, Lluis Padró, and Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. In *LREC*.

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation Conference*, volume 6, pages 417–422.

Joan Lucariello. 1994. Situational irony: A concept of events gone awry. *Journal of Experimental Psychology: General*, 123(2):129.

10.12871/clicit2014219

Self-Evaluating Workflow for Language-Independent Sentiment Analysis

Arseni Anisimovich
Minsk State Linguistic University, Minsk, Belarus[1]
WorkFusion Inc., New York, USA[2]
arseni.anisimovich@gmail.com[1]
aanisimovich@workfusion.[2]

Abstract

English. This paper describes a generic framework that relies on extra-linguistic features of text as well as on its content to perform sentiment analysis in four different dimensions. Routine described in the paper allows not only extraction of opinion mining data but also describes a framework for continuous relearning of Support Vector Machines classifiers in order to improve classification results when dataset size is increased or new parameters of classifier are found to be of better quality.

Italiano. *Questo articolo descrive una tecnica generale che si basa su caratteristiche extra-linguistiche del testo, e anche sul suo contenuto, allo scopo di eseguire una sentiment analysis in quattro dimensioni. Questo procedimento non solo permette l'estrazione dei dati di sentiment analysis, ma descrive anche un algoritmo di ri-apprendimento continuo con support vector machines (particolarmente utile nei casi in cui ci sono ulteriori esempi o nuovi parametri che migliorano la qualità dell'analisi).*

1 Introduction

The rise of new media especially social ones have brought absolutely new source of up-to-date information on different topics that can be exploited in different tasks. One of such tasks is opinion mining or sentiment analysis that could bring vital information to many researchers including, but not limited to sociologists, campaigners, and marketing analysts.

Sentiment analysis of English texts has drawn scholars' attention about a decade ago (Turney, 2002; Pang et al., 2002) and provided basic experimental data and directions of research for scientific community. That resulted in annual shared tasks and conferences that bring attention to the problem and raise the bar for the state-of-the-art approaches on a regular basis.

However, the information to be analyzed in modern world does not include sole English texts. That fact has inspired raising interest in developing mechanisms for sentiment analysis of texts in languages other than English (Basile et al., 2014). While some scholars propose the focus on leveraging resources from languages with more data (Mihalcea et al., 2007), this paper describes a generic approach in sentiment analysis that can be applied to any collection of labelled data without preliminary linguistic work.

2 System Description

Sentiment analysis, as the task that this paper is aimed to solve, is a basic binary classification problem when treating each of sub-tasks (Positive and Negative Polarity, Subjectivity and Irony) as a separate problem.

Recent researches prove that in sentiment analysis as a classification task, Support Vector Machines (SVM) classifiers perform with a decent quality (Mullen, 2004), (Gamon, 2004). LibSVM (Chang, 2011) was used as an algorithmic implementation of SVMs.

Since libSVM comes with several Support Vector Machines types and several kernels, the workflow was set up to train all applicable classifiers with a ranging parameters to automatically find the best configurations for every classification task.

SVM's possibility to train a stable classifier on a limited set of labeled data has been of a huge help because of variable proportion of positive and negative examples of a class in each subtasks:

	Pos. Example	Neg. Example
Subjectivity	2804 (70.68%)	1163 (29.32%)
PolPositive	1132 (28.54%)	2835 (71.46%)
PolNegative	1729 (43.58%)	2238 (56.42%)
Irony	498 (12.55%)	3469 (87.46%)

Table 1. Amount of examples per subtask.

Despite the fact that positive and negative example ratio is different per task, training set was unified for every subtask as well as the features selection. The main ranging parameters were SVM parameters and feature frequency threshold.

Since results were only reported for constrained run, there was no external information used in the feature set. However, several simple text transformations were performed to facilitate classifier training basing on extra-linguistic knowledge.

2.1 Feature Selection

The assumption that the set of features is similar in all subtasks was made thus eliminating the need for several training set generation procedures. However, several transformations of raw tweet text were performed.

Firstly, all URLs were converted to a single word-marker '*url*' because of insignificancy of link address. Then, the presumption that some links bring more personalized information was token, and the URLs were classified into two groups: Long URLs and Shortened URLs. The former is a link in an unconstrained format peculiar to a specific website while the latter is provided by third-party service (e.g. Google URL Shortener[1], Bitly[2], or Twitter's internal service[3]).

The reason behind that transformation is that when an application (either way on a mobile device or in a browser) posts a link, it usually converts a given URL in short format (in order to save the space in a 140-symbol message), but, as the research of training dataset has shown, when a news agency posts a link, it usually posts it as-is, without any shortening service. Since the information whether the tweet belongs to an individual or to an organization is a valuable feature, this transformation was applied for every tweet and gave 2% average increase in terms of both precision and recall.

Another important transformation of dataset was to turn all the variety of smileys into information. From all the smileys only two categories were selected: those representing a sad emotion and those representing a happy emotion, since polarity task had only two dimensions and variety of emotions that can be represented using smileys is convertible to these two subsets.

Except of described transformations, size of tweet relative to maximum size of tweet in training dataset (in bytes) was added to raw text as well as quotation markers, uncertainty or fragmentary text markers (for example three dots), re-tweet markers, hashtag markers, and Twitter picture (pic.twitter.com) markers in order to catch all the information that not only exists outside of the language, but is a distinctive feature of modern Internet communication and its implementation (Twitter as a platform and its client applications as instruments). Described transformations may be applied to any tweet in any language and still will produce comparable amount of training information.

2.2 Vector Normalization

Since SVM is a vector-based classifier and requires a vector of values as input for both training and classification procedure, a binary vector for each document was built using token occurrence as a '1' value and token absence as '0'. Token is understood as a sequence of non-whitespace characters.

This approach is usual to SVM feature generation, however it lacks the information about number of occurrences of a token in the text, and if in the case of stop word this information will not give any classification weight at all, quantity of emotion markers or picture amount in the text are priceless information which might be the straw that may break the back of misclassification camel.

Since the value of every token was only 0 or 1, in the described approach token occurrence in a document was scaled with maximum token occurrence in the training dataset thus turning possible values of a single feature from binary 0/1

[1] https://goo.gl/
[2] https://bitly.com/
[3] http://t.co/

vector into vector of values 0..1 thus saving the information for classifier to train on.

SVM's vector nature was a huge gain when compared to probability-based classifiers, since if one class tends to have less token occurrences and in testing set there is even smaller amount of those, SVM will not turn that feature into non-relevant, but will do its best to correctly classify example by comparing incoming vector against trained hyper plane.

2.3 Feature Pruning

As it was mentioned earlier, amount of positive and negative examples for each dimension of sentiment analysis varies a lot, leading to great feature imbalance. One of the approaches that can be used to eliminate negative impact on sentiment analysis quality is feature frequency limitation mechanism that excludes from training and testing vector those features that occur less than a predefined threshold.

Despite the fact that there are approaches that exclude features on the basis of discriminative function pruning analysis (DFPA)(Mao, 2004) this paper sticks to examinations of options to select most corresponding minimal feature frequency suitable for each subtask. Optimal parameters vary greatly, for example:

	PolNegative Precision	**PolPositive Precision**
FeatFreq: 15	35,46%	57,85%
FeatFreq: 4	38,82%	49,32%

Table 2. Precision changes over feature frequency parameter selection.

Automatic routine of choosing best parameters allows not only find best values for current task with current dataset, but also, if a researcher has access to continually growing dataset, existing models may be retrained in background with dataset growth and achieve better quality over new data.

2.4 Experimental Workflow

As it was said above, initial dataset for solving each of four subtasks is the same and when it comes into the system, training procedure begins from same starting point. Baseline of precision and recall is set using one-rule classifier (presuming that all examples should be classified as the majority of examples in training set).

Baseline is used to exclude those combinations of SVM types and kernel types that bring results worse than baseline (however, in this particular task, it never occurred and all applicable SVM classifiers were training all at once).

To eliminate the threat of biased testing set ten-fold cross-validation is used on every set of parameters during evaluation of classifier. Average of precisions and recalls for each cross-validation run is then used to rank set of parameters as most or least applicable to a given classification task.

Set of classifier parameters varies from SVM type and kernel type, and the only common parameter is feature frequency threshold. Experiments have shown that for the SENTIPOLC-2014 task for described approach following feature frequencies limits bring best results:

Irony	3
Subjectivity	15
PolPositive	3
PolNegative	7

Table 3. Feature frequencies thresholds per subtask.

These results correlate with common sense knowledge since both irony and positive attitude can be expressed in many ways and negative attitude, despite being expressed more often than positive attitude, lacks that variety of words to use. Limitations of Twitter message size and Internet slang provides a set of shorthands to express subjectivity and stay in the margins of tweet.

Different SVMs also train with different parameters specific to an algorithm, for example for linear SVMs the parameter *C* (cost parameter) was ranged from default 1 up to 100, for nu-SVC *v* (nu) parameter was ranged from 0.01 up to 0.45 . Best parameters are selected for all the SVM and kernel types.

In the last step framework chooses best combination of feature frequency, SVM type and kernel type and trains final model on whole dataset to have a 'production' model that will be used to rank against testing data. In the SENTIPOLC-2014 task following parameters were chosen for each subtask:

Subtask	FeatFreq	Classifier (type/kernel)
Irony	3	c-SVC, linear (c=11)
Subjectivity	15	c-SVC, linear (c=11)
PolPositive	3	c-SVC, linear (c=9)
PolNegative	7	ν-SVC, linear (ν=0.43)

Table 4. Parameters of SVM classifiers.

All subtasks except for negative polarity were ranked using F1-measure while negative polarity was ranked using classification precision since basically, any F1-measure best classifier was one-rule classifier totally missing positive examples of negative polarity.

3 Conclusion

Described system didn't take first places in any constrained run task in SENTIPOLC-2014 shared task. However, resulting scores correlated with those obtained in cross-validation of 'production' classifiers while being 5-10% lower than development ones:

Subtask	Expected	Real	Top
Subjectivity 7/9	0.6545	0.5825	0.7140
Polarity 6/11	0.6812	0.6026	0.6771
Irony 3/7	0.5828	0.5394	0.5901

Table 5. Expected results with rankings.

Nonetheless, the approach presented in this paper has proven itself valid to be used against Twitter messages without any preliminary linguistic work. Features were independent from language of a tweet and all text transformations may be applied to a message in any language.

Described approach, unfortunately, lacks the information about syntactic structure of text of the tweet which may be eliminated or at least leveled with the help of a standard syntactic parser that should provide a uniform representation of syntactic structure for any language given, for example, dependency grammar tree.

In unconstrained run, there is a point of constant update of a training set using crowd sourcing platforms, which can provide data with high quality using initial training set not only as a classifier training set, but also as an example to teach crowd workers and maintain their quality as described in (Lease, 2011). That will give not only more complete dataset, but also will provide sources for relearning the classifier on new data that may reflect changes in the Internet slang that may occur in a split second.

References

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*.

Chih-Chung Chang, and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*.

Michael Gamon. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. *Proceedings of the 20th international conference on Computational Linguistics*. ACL.

Matthew Lease. 2011. On Quality Control and Machine Learning in Crowdsourcing. *Human Computation*.

Mao, K.Z. 2004. Feature subset selection for support vector machines through discriminative function pruning analysis. *Systems, Man, and Cybernetics, Part B: Cybernetics*. Vol. 34, Issue 1. IEEE.

Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning Multilingual Subjective Language via Cross-Lingual Projections. *Proceedings of the Association for Computational Linguistics (ACL)*. pp. 976–983.

Tony Mullen, and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. *EMNLP*. Vol. 4.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 79–86.

Peter Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the Association for Computational Linguistics.* pp. 417–424.

10.12871/clicit2014220

EVALITA 2014: Emotion Recognition Task (ERT)

Antonio Origlia
University "Federico II", Napoli, Italy
antonio.origlia@unina.it

Vincenzo Galatà
ISTC-CNR, UOS Padova, Italy
Free University of Bozen-Bolzano, Italy
vincenzo.galata@pd.istc.cnr.it

Abstract

English. In this report, we describe the EVALITA 2014 Emotion Recognition Task (ERT). Specifically, we describe the datasets, the evaluation procedure and we summarize the results obtained by the proposed systems. On this basis we provide our view on the current state of emotion recognition systems for Italian, whose development appears to be severely slowed down by the type of data available nowadays.

Italiano. *In questo report, descriviamo il task EVALITA 2014 dedicato al riconoscimento di emozioni (ERT). In particolare, descriviamo i set di dati utilizzati, la procedura di valutazione e riassumiamo i risultati ottenuti dai sistemi proposti. Su questa base, descriveremo la nostra posizione sullo stato attuale dei sistemi per il riconoscimento di emozioni per l'Italiano, il cui sviluppo sembra essere fortemente rallentato dal tipo di dati disponibili attualmente.*

1 Introduction

After the Interspeech 2009 Emotion Challenge (Schuller et al., 2009) and the Interspeech 2010 Paralinguistics Challenge (Schuller et al., 2010), the EVALITA Emotion Recognition task (ERT) represents the first evaluation campaign specifically dedicated to Italian Emotional speech. Unlike the two Interspeech challenges, we move here the first steps for Italian by using acted emotional speech collected according to Ekman's classification model (Ekman, 1992) as this is, so far, the only type of speech material we have knowledge. In this task, we aimed at evaluating the performance of automatic emotion recognition systems and to investigate two main topics, covered by two different subtasks:

- cross language, open database task
- Italian only, closed database task

First of all, we wanted to estimate the performance that could be obtained on Italian using emotional speech corpora in other languages. We also wanted to verify to what extent it would have been possible to build a model for emotional speech starting from a single, professional, speaker portraying the discrete set of emotions defined by Ekman (1992) (anger, disgust, fear, joy, sadness, surprise, and neutral).

In this first evaluation of emotional speech recognition systems on Italian, the material we use is composed of acted speech elicited by means of a narrative task. The material is extracted from two emotional speech corpora containing similar material and sharing basic characteristics:

- the E-Carini corpus
- the €motion corpus

Concerning the second subtask, the goal of the evaluation was to establish how much information could be extracted from material coming from a single, professional source of information whose explicit task is to portray emotions and obtain models capable of generalizing to unseen subjects.

2 Datasets

For both development and training sets, *.wav files were provided along with their Praat *.TextGrid file containing a word level (wrd) annotation carried out by means of forced alignment. Pauses in the *.TextGrid file are labelled as ".pau". The material consists of PCM encoded WAV files (16000Hz).

2.1 Development set: the €motion corpus

Participants were provided with a development set taken from the yet unpublished €motion corpus (Galatà, 2010) to obtain reference results for the test material during the system preparation time. The material extracted from €motion consists of the Italian carrier sentence "Non è possibile. Non ci posso credere." (*It can't be. I can't believe it.*), recorded by one professional actor according to 4 instructions (or recording modes) as follows:

- Mode A: after a private reading, read again the six scenarios with sense and in a natural and spontaneous way;
- Mode B: read the text once more with sense and in a natural and spontaneous way considering the desired emotion letting himself personally get involved in the story proposed in the text;
- Mode C: repeat the carrier sentence according to the requested emotion and to the scenario proposed in each text;
- Neutral mode: simply read a list of sentences (containing the carrier sentence).

Following the above described elicitation procedure, the 40 sentences were provided as development set:

- Mode A: 6 productions (1 per emotion);
- Mode B: 6 productions (1 per emotion);
- Mode C: 24 productions (4 per emotion);
- Neutral mode: 4 neutral productions.

The file name structure for this data set provides information on the way the sentence has been collected as well as the discrete emotion label assigned and intended for its production. Given the file name it_ang_a_mt_c1 as example, the file name provides the following information:

- Language: it;
- Intended emotion: 6+1 discrete emotion labels (eg. ang, sur, joy, fea, sad, dis, neu);
- Type of subject: a (actor);
- Subjects name: mt;
- The recording mode: a, b or c (for the neutral mode this slot is left out);
- Occurrence number: 1, 2, 3 or 4.

2.2 Training set: the E-Carini corpus

The material provided for the E-Carini corpus (Avesani et al., 2004; Tesser et al., 2004; Tesser et al., 2005), consists of a reading by a professional actor of the short story "Il Colombre" by Dino Buzzati. The novel is read and acted according to the different discrete emotion labels provided. The novel is split in 47 paragraphs (from *par01* to *par47* in the file name) and stored in different folder (one for each emotion). This training set provided for the *closed database* task consisted of 1 hour and 17 minutes of speech.

2.3 The test set

All the participants were provided with the test set consisting of emotional productions by 5 actors with the same characteristics as in the *development set* above described. For each emotion, 30 stimuli were included in the test set. In order to allow speaker dependent system training, 4 neutral productions were provided for each speaker in the test set.

All the file names provided for the *test set*, apart from the neutral ones, were masked: the subject ID was, however, available to the participants, while the target emotion was kept hidden. The format given to the files contained the subjects name followed by a three digits random number (eg. as_108). Neutral files followed the format provided with the *development set* files.

3 Evaluation measure

Typically, the objective measure chosen for an emotion classification task would be the F-measure. However, as in this case, the sample accuracy (percentage of correctly classified instances) is used. Since the test set here distributed contains the same number of examples for each class, there is no influence to take into account on the side of data distribution and the sample accuracy results in a better choice.

3.1 Baseline

For the emotion recognition system baseline, we used the features set obtained with the OpenSMILE package (Eyben et al., 2013) in the configuration used for the Interspeech 2010 Paralinguis-

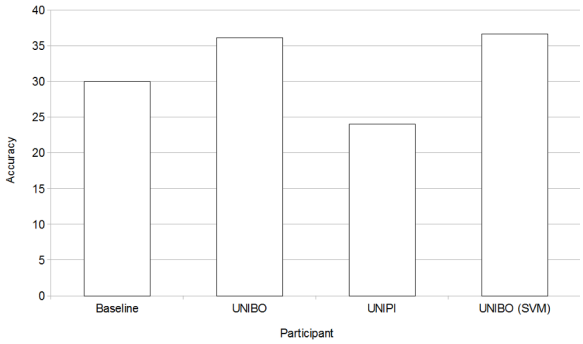


Figure 1: Summary of the submitted results. We also report the experiment provided by UNIBO with an SVM trained with finer parameter optimization than the one used as a baseline.

tics Challenge (Schuller et al., 2010). The LibSVM (Chang and Lin, 2011) implementation of Support Vector Machines (SVM) was trained on this data using an RBF kernel and a basic strategy to optimize the γ and C parameters (grid search between 0 and 2 with 0.4 grid step for both). The obtained classifier reached an accuracy of 30% on the test set.

4 Participation and results

Before receiving the material, all participants were asked to sign an End User License Agreement (EULA). Four participants downloaded the datasets after publication on the EVALITA website.

However, after receiving the test material, only two participants submitted the final test results for the "closed database" subtask and no one for the "open database" subtask. A system from the University of Bologna (UNIBO) and the University of Pisa (UNIPI) were proposed. Results were submitted to the organizers as a two columns *.csv file: the first column containing the file name and the second column the label assigned by the proposed system (eg. as_100, ang; eo_116, fea; etc.).

After the results submission, the participants were provided with a rename table mapping the masked file names on the original ones in order to let them replicate the evaluation results. In the following subsections we summarize the proposed approaches, while in Figure 1 we show the graphical comparison among the approaches with their respective recognition accuracies.

4.1 UNIBO

The system presented by UNIBO performed emotion recognition by means of a Kernel Quantum Classifier, a new general-purpose classifier based on quantum probability theory. The system is trained on the same feature set used for the baseline. The system reached a performance of 36.11% recognition accuracy, which is the highest result obtained in the ERT.

4.2 UNIPI

The system presented by UNIPI used an Echo State Network (Jaeger and Haas, 2004) to perform emotion classification. The system has the peculiarity of receiving, as input, directly the sound waveform, without performing features extraction. Neutral speech productions for each speaker were used to obtain waveform normalization constants for each speaker. Using the proposed approach, a recognition accuracy of 24% was obtained on the test set.

5 Discussion

The results obtained in the ERT task highlight an important problem for emotion recognition speech in Italian concerning the available material. While corpora containing Italian acted emotional productions have been successfully used for emotional speech synthesis in the past (this is the case of the E-Carini corpus), it appears it is not straightforward to transfer the model built on one professional actor portraying a set of specific emotions on other subjects, even if they are professional actors too. As a consequence, we believe that the type of emotional speech data available nowadays is inadequate to train emotion recognition systems for Italian. The reason for this inadequacy is mainly due to the difference between the type of data collected so far for Italian and the data that have been collected in other countries (mostly English speaking). For Italian, other than the E-Carini and the €motion corpus, to our knowledge only the EMOVO corpus (Iadarola, 2007; Costantini et al., 2014) is available. This dataset, as the ones here adopted, also contains acted read speech classified using Ekmans schema. Outside Italy, on the contrary, the scientific community appears to be oriented towards more spontaneous speech, mostly elicited through dialogue with artificial agents in a Wizard of Oz setup and annotated with both emotional classes and with continuous

measures as done, for example, in the SEMAINE corpus (McKeown et al., 2010). As a matter of fact, the latest international challenges on emotion recognition are evaluated on the capability of automatic systems to track continuous values over the entire utterance (regression), as opposed to recognizing a single class over a full sentence (classification).

In conclusion, the result of the EVALITA 2014 ERT task seems to highlight that the type of data available in Italian emotional speech corpora is outdated at least for the emotion recognition task. Two problems are, in our opinion, important for the Italian community to tackle. First of all, we have observed that it is not straightforward to transfer the knowledge acquired by modelling a single professional source to other professional sources even in the case of read speech in silent conditions with a neutral speech basis available. This indicates that it is necessary for the Italian community working on emotional speech recognition to move away from this kind of data and collect more spontaneous data.

The second problem lies in data annotation. On an international level, automatic classification according to Ekmans basic emotions has been abandoned in favour of dimensional models as proposed, for example, by Mehrabian (1996). We believe it is necessary for the Italian community to move forward in this sense too as the global attention appears to be focused on dimensional annotations.

Acknowledgments

This work has been funded by the European Community and the Italian Ministry of University and Research (MIUR) and EU under the PON OR.C.HE.S.T.R.A. project. Vincenzo Galatà's work has been supported by WIKIMEMO.IT (The Portal of Italian Language and Culture, FIRB-MIUR Project, RBNE078K93).

References

Cinzia Avesani, Piero Cosi, Elisabetta Fauri, Roberto Gretter, Nadia Mana, Silvia Rocchi, Franca Rossi, and Fabio Tesser. 2004. Definizione ed annotazione prosodica di un database di parlato-letto usando il formalismo ToBI. In *Il parlato italiano*, pages 1–14.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Giovanni Costantini, Iacopo Iaderola, Andrea Paoloni, and Massimiliano Todisco. 2014. EMOVO corpus: an Italian emotional speech database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA).

Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200.

Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. 2013. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838.

Vincenzo Galatà. 2010. *Production and perception of vocal emotions: a cross-linguistic and cross-cultural study*. Ph.D. thesis, University of Calabria - Italy.

Iacopo Iadarola. 2007. EMOVO: database di parlato emotivo per l'italiano. In *Atti del 4 Convegno Nazionale dell'Associazione Italiana Scienze della Voce (AISV)*.

Herbert Jaeger and Harald Haas. 2004. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80.

Gary McKeown, Michel Franois Valstar, Roderick Cowie, and Maja Pantic. 2010. The semaine corpus of emotionally coloured character interactions. In *Proc. of ICME*, pages 1079–1084.

Albert Mehrabian. 1996. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology: Developmental, Learning, Personality, Social*, 14:261–292.

Björn Schuller, Stefan Steidl, and Anton Batliner. 2009. The INTERSPEECH 2009 emotion challenge. In *Proc. of Interspeech*, pages 312–315. ISCA.

Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian A Müller, and Shrikanth S Narayanan. 2010. The INTERSPEECH 2010 paralinguistic challenge. In *Proc. of Interspeech*, pages 2794–2797.

Fabio Tesser, Piero Cosi, Carlo Drioli, and Graziano Tisato. 2004. Modelli prosodici emotivi per la sintesi dell'italiano. *Proc. of AISV 2004*.

Fabio Tesser, Piero Cosi, Carlo Drioli, and Graziano Tisato. 2005. Emotional Festival - Mbrola TTS synthesis. *Interspeech 2005*, pages 505–508.

10.12871/clicit2014221

A Preliminary Application of Echo State Networks to Emotion Recognition

Claudio Gallicchio
Department of Computer Science
University of Pisa
Largo B. Pontecorvo 3
56127 Pisa, Italy
gallicch@di.unipi.it

Alessio Micheli
Department of Computer Science
University of Pisa
Largo B. Pontecorvo 3
56127 Pisa, Italy
micheli@di.unipi.it

Abstract

English. This report investigates a preliminary application of Echo State Networks (ESNs) to the problem of automatic emotion recognition from speech. In the proposed approach, speech waveform signals are directly used as input time series for the ESN models, trained on a multi-classification task over a discrete set of emotions. Within the scopes of the Emotion Recognition Task of the Evalita 2014 competition, the performance of the proposed model is assessed by considering two emotional Italian speech corpora, namely the E-Carini corpus and the €motion corpus. Promising results show that the proposed system is able to achieve a very good performance in recognizing emotions from speech uttered by a speaker on which it has already been trained, whereas generalization of the predictions to speech uttered by unseen subjects is still challenging.

Italiano. *Questo documento esamina l'applicazione preliminare delle Echo Stato Networks (ESN) per il problema del riconoscimento automatico delle emozioni dal parlato. Nell'approccio proposto, i segnali che rappresentano la forma d'onda del parlato sono usati direttamente come serie temporali di ingresso per i modelli ESN, addestrati su un compito di multi-classificazione su un insieme discreto di emozioni. Entro gli ambiti della Emotion Recognition Task della competizione Evalita 2014, la performance del modello proposto viene valutata considerando due corpora di dati emotivi in lingua Italiana, ovvero il corpus E-Carini e il corpus €motion. I risultati ottenuti sono promettenti e mostrano che il sistema proposto è in grado di raggiungere una buona prestazione nel riconoscimento di emozioni a partire dalle parole pronunciate da un utente sul quale il sistema è stato già addestrato, mentre la generalizzazione delle predizioni per le frasi pronunciate da soggetti mai visti in fase di addestramento rappresenta ancora un aspetto ambizioso.*

1 Introduction

The possibility of recognizing human emotions from uttered speech is a recent interesting area of research, with a wide range of potential applications in the field of human-machine interactions. One of the most prominent aspects of recent systems for emotion recognition from speech relates to the choice of proper features that should be extracted from the waveform signals. Popular choices for such features are continuous features (Lee and Narayanan, 2005), such as pitch-related features or energy-related features, or spectral based features, such as linear predictor coefficients (Rabiner and Schafer, 1978) or Mel-frequency cepstrum coefficients (Bou-Ghazale and Hansen, 2000).

Within the scopes of the Evalita 2014 competition, this report describes a preliminary investigation of the application of Echo State Networks (ESNs) (Jaeger and Haas, 2004) to the problem of identifying speakers' emotions from a discrete set, namely anger, disgust, fear, joy, sadness and surprise. We adopt the paradigm of Reservoir Computation (Lukosevicius and Jaeger, 2009), which represents a state-of-the-art approach for efficient learning in time-series domains, within the class of Recurrent Neural Networks, naturally suitable for treating sequential/temporal information. As such, in our proposed approach, the waveform sig-

nals representing speech are directly used as input for the emotion recognition system, allowing to avoid the need for domain-specific feature extraction from waveform signals. In order to assess the generalization performance of the proposed emotion recognition system, we take into consideration a homogeneous experimental setting and a heterogeneous experimental setting. In the homogeneous setting, the performance of the recognition system is assessed on sentences uttered by the same speaker on which the system has been trained, while in the heterogeneous setting, the performance is assessed on sentences pronounced by unseen subjects during the training process.

2 Description of the System

We took into consideration data coming from two emotional Italian speech corpora, namely the E-Carini corpus (Tesser et al., 2005; Avesani et al., 2004) and the €motion corpus (Galatà, 2010). Each corpus contains waveform signals representing sentences spoken by a single user, see the task report (this volume) for further details. Such data was then organized into two datasets, one for each corpus, segmenting sentences into words, based on the available information. Our emotion recognition system directly uses the sounds waveform of spoken words as input time-series for the neural network model, avoiding the use of feature extraction for speech representation. The only preprocessing step consists in normalizing the input signals to zero mean and unitary standard deviation, using the data pertaining to the extra neutral emotion class for computing the normalization constants, independently for each speaker.

The two resulting datasets were used to organize two multi-classification task for emotion recognition: a homogeneous task and a heterogeneous task. The homogeneous task includes only the E-Carini corpus dataset, and is designed for assessing the ability of the emotion recognition system to detect human emotions pertaining to a single speaker. Indeed, training and test set for the homogeneous task contain sequences pertaining to the same speaker (test set represents $\approx 30\%$ of the available data). The heterogeneous task includes both the E-Carini corpus and the €motion corpus, and is designed to evaluate the generalization ability of the emotion recognition system when trained on data pertaining to one speaker and tested on data pertaining to a different speaker. In the case of the heterogeneous task, the training set contains data from the E-Carini corpus, while the test set contains data from the €motion corpus. For both the homogeneous and the heterogeneous tasks, the training set was balanced over the class of possible emotions.

Emotion classification is performed by using ESN, which implement discrete-time non-linear dynamical systems. From an architectural perspective, an ESN is made up of a recurrent *reservoir* component, and a feed-forward *readout* component. In particular, the reservoir part updates a state vector which provides the network with a non-linear dynamic memory of the past input history. This allows the state dynamics to be influenced by a portion of the input history which is not restricted to a fixed-size temporal window, enabling to capture longer term input-output relationships. In the context of the specific application under consideration, it is worth noticing that the role of the reservoir consists in directly encoding the temporal sequences of the waveform signals into a fixed-size state (feature) vector, allowing to avoid the need for the extraction of specific features from the uttered sentences. The basic architecture of an ESN includes an input layer with N_U units, a non-linear, recurrent and sparsely connected reservoir layer with N_R units, and a linear, feed-forward readout layer with N_Y units. In particular, for our application we use $N_U = 1$ and $N_Y = 6$, where each one of the output dimensions corresponds to one of the emotional classes considered. In this paper we take into consideration the leaky integrator ESN (LI-ESN) (Jaeger et al., 2007), which is a variant of the standard ESN model, with state dynamics particularly suited for representing the history of slowly changing input signals.

State dynamics of the ESNs follow the word by word segmentation organization considered in the datasets. Accordingly, for each word w, at each time step t, the reservoir computes a state $\mathbf{x}_w(t) \in \mathbb{R}^{N_R}$ according to the equation:

$$\begin{aligned}\mathbf{x}_w(t) = \quad & (1-a)\mathbf{x}_w(t-1) + \\ & af(\mathbf{W}_{in}\mathbf{u}_w(t) + \hat{\mathbf{W}}\mathbf{x}_w(t-1))\end{aligned} \tag{1}$$

where $\mathbf{u}_w(t)$ is the input at time-step t, $\mathbf{W}_{in}$ is the input-to-reservoir weight matrix, $\mathbf{W}$ is the recurrent reservoir weight matrix, $a \in [0, 1]$ is a leaking rate parameter, f is an element-wise applied activation function (we use $tanh$), and a zero vector

is used for state initialization. After the last time step for word w has been considered, a mean state mapping function is applied, according to:

$$\mathcal{X}(w) = \frac{1}{length(w)} \sum_{t=1}^{length(w)} \mathbf{x}_w(t) \qquad (2)$$

where $length(w)$ is the number of time steps covered by the sentence w. For further information about state mapping functions in general, and mean state mapping in particular, the reader is referred to (Gallicchio and Micheli, 2013).

The classification output is computed by the readout component of the ESN, which linearly combines the output of the state mapping function, according to the equation:

$$\mathbf{y}(w) = \mathbf{W}_{out}\mathcal{X}(w) \qquad (3)$$

where $\mathbf{W}_{out}$ is a reservoir-to-readout weight matrix. The emotional class for each word is set to the class corresponding to the element with the highest activation in the output vector. The final classification of a sentence is computed by a voting process, according to which each sentence is classified as belonging to the emotional class which is more represented among the words that compose that sentence.

Training in ESNs is restricted to only the readout component, i.e. only the weight values in matrix $\mathbf{W}_{out}$ are adapted, while elements in $\mathbf{W}_{in}$ and $\mathbf{W}$ are initialized in order to satisfy the conditions of the *echo state property* (Jaeger and Haas, 2004) and then are left untrained. In practical applications, such initialization process typically consists in a random initialization (from a uniform distribution) of weight values in matrices $\mathbf{W}_{in}$ and $\mathbf{W}$, after which matrix $\mathbf{W}$ is scaled such that its spectral radius $\rho(\mathbf{W})$ is less than 1, see (Jaeger, 2001) and (Gallicchio and Micheli, 2011) for details.

3 Results

In our experiments we considered ESNs with reservoir dimension $N_R \in \{100, 200\}$, 10% of reservoir units connectivity, spectral radius $\rho = 0.999$ and leaky parameter $\alpha = 0.01$. For every reservoir hyper-parametrization, results were averaged over a number of 10 reservoir guesses. The readout part of the ESNs was trained using pseudo-inversion and ridge regression with regularization parameter $\lambda \in \{10^j | j = -5, -4, -3, -2, -1, 0, 1, 2, 3\}$. Reservoir dimension and readout regularization were chosen on a validation set (with size of $\approx 30\%$ of the training set size), according to a hold out cross validation scheme for model selection.

The performance of the emotion recognition is assessed by measuring the accuracy for the multi-classification task, i.e. the ratio between the number of correctly classified sentences and the total number of sequences. Average training and test accuracy obtained on both the homogeneous and heterogeneous tasks are reported in Table 3.

Task	**Training**	**Test**
homogeneous	$0.86(\pm 0.01)$	$0.82(\pm 0.01)$
heterogeneous	$0.91(\pm 0.02)$	$0.27(\pm 0.03)$

Table 1: Average training and test performance accuracy achieved by ESNs on the homogeneous task and on the heterogeneous task.

For the sake of performance comparison, notice that the accuracy achieved by a chance-null model is 0.17 on both the tasks. The averaged accuracy achieved on the test set of the homogeneous task is 0.82, which is comparable with literature results on emotion recognition from speech in homogeneous training-test condition (Ayadi et al., 2011). The averaged accuracy achieved on the test set of the heterogeneous task is 0.27. Note that, although such performance is far from the one achieved on the homogeneous task, it is still definitely beyond the performance of the null model. The result achieved by the system trained on the heterogeneous case on the full test set of the Evalita 2014 competition, comprising data from 5 different unseen speakers, is 0.24.

4 Discussion

In this report we have described a preliminary application of ESNs to the problem of recognizing human emotions from speech. The proposed emotion recognition system directly uses as input the time series of the waveform signals corresponding to the uttered sentences, avoiding the need for a specific feature extraction process. Two experimental settings have been considered, with training and test data pertaining to sequences pronounced by the same speaker (homogeneous setting) or not (heterogeneous setting). Performance results achieved by ESNs are promising. In particular, a very good predictive performance is ob-

tained when the system is assessed considering unseen sentences pronounced by a speaker on which the system has already been trained. On the other hand, the generalization of the emotion predictions to speech uttered by speakers on which the system has not been trained still remains a challenging aspect. Overall, given the characteristics of efficiency and simplicity of the proposed approach, and in view of a possible integration with domain-specific techniques for the multi-speaker case, we believe that the proposed system can represent an interesting contribution for the design of tools in the area emotional speech processing.

References

Cinzia Avesani, Piero Cosi, Elisabetta Fauri, Roberto Gretter, Nadia Mana, Silvia Rocchi, Franca Rossi, and Fabio Tesser. 2004. Definizione ed annotazione prosodica di un database di parlato-letto usando il formalismo tobi. In *Il parlato Italiano*, pages 1–14.

Moataz El Ayadi, Mohamed Kamel, and Fakhri Karray. 2011. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587.

Sahar E. Bou-Ghazale and John Hansen. 2000. A comparative study of traditional and newly proposed features for recognition of speech under stress. *Speech and Audio Processing, IEEE Transactions on*, 8(4):429–442.

Vincenzo Galatà. 2010. Production and perception of vocal emotions: a cross-linguistic and cross-cultural study. PhD Thesis, University of Calabria, Italy, (unpublished).

Claudio Gallicchio and Alessio Micheli. 2011. Architectural and markovian factors of echo state networks. *Neural Networks*, 24(5):440 – 456.

Claudio Gallicchio and Alessio Micheli. 2013. Tree echo state networks. *Neurocomputing*, 101:319–337.

Herbert Jaeger and Harald Haas. 2004. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80.

Herbert Jaeger, Mantas Lukosevicius, Dan Popovici, and Udo Siewert. 2007. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–352.

Herbert Jaeger. 2001. The "echo state" approach to analysing and training recurrent neural networks. Technical report, GMD.

Chul Min Lee and Shrikanth Narayanan. 2005. Toward detecting emotions in spoken dialogs. *Speech and Audio Processing, IEEE Transactions on*, 13(2):293–303.

Mantas Lukosevicius and Herbert Jaeger. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

Lawrence Rabiner and Ronald Schafer. 1978. *Digital Processing of Speech Signals*. Pearson Education.

Fabio Tesser, Piero Cosi, Carlo Drioli, and Graziano Tisato. 2005. Emotional festival-mbrola tts synthesis. In *INTERSPEECH*, pages 505–508.

10.12871/clicit2014222

Emotion Recognition with a Kernel Quantum Classifier

Fabio Tamburini
FICLIT - University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

English. This paper presents the application of a Kernel Quantum Classifier, a new general-purpose classifier based on quantum probability theory, in the domain of emotion recognition. It participates to the EVALITA 2014 Emotion Recognition Challenge exhibiting relatively good results and ranking at the first place in the challenge.

Italiano. *Questo contributo presenta l'applicazione di un classificatore quantistico basato su kernel, un nuovo classificatore basato sulla teoria della probabilità quantistica, nel dominio del riconoscimento delle emozioni. Ha partecipato alla campagna di valutazione sul riconoscimento delle emozioni nell'ambito di EVALITA 2014 ottenendo buoni risultati e classificandosi al primo posto.*

1 Introduction

Quantum Mechanics Theory (QMT) is one of the most successful theory in modern science. Despite its ability to properly describe most natural phenomena in the physics realm, the attempts to prove its effectiveness in other domains remain quite limited.

This paper presents the application of a Kernel Quantum Classifier, a new general-purpose classifier based on quantum probability theory, in the domain of emotion recognition.

With regard to this specific evaluation challenge, we did not develop any particular technique tailored to emotion recognition, but we applied a "brute force" approach to this problem as described, for example, in (Schuller *et al.*, 2009). A very large set of general acoustic features has been automatically extracted from speech waveforms and the emotion detection task has been put totally in charge of the classifier.

In section 2 we will describe the proposed classifier, in section 3 the evaluation results will be analysed comparing them with the results obtained using a state-of-the-art classifier applied to the same task and in section 4 we will draw some provisional conclusions.

2 System description

2.1 Quantum Probability Theory

A *quantum state* denotes an unobservable distribution which gives rise to various observable physical quantities (Yeang, 2010). Mathematically it is a vector in a complex Hilbert space. It can be written in Dirac notation as $|\psi\rangle = \sum_1^n \lambda_j |e_j\rangle$where λ_j are complex numbers and the $|e_j\rangle$ are the basis of the Hilbert space ($|.\rangle$ is a column vector, or a *ket*, while $\langle .|$ is a row vector, or a *bra*). Using this notation the inner product between two state vectors can be expressed as $\langle\psi|\phi\rangle$ and the outer product as $|\psi\rangle\langle\phi|$.

$|\psi\rangle$ is not directly observable but can be probed through measurements. The probability of observing the elementary event $|e_j\rangle$ is $|\langle e_j|\psi\rangle|^2 = |\lambda_j|^2$ and the probability of $|\psi\rangle$ collapsing on $|e_j\rangle$ is $P(e_j) = |\lambda_j|^2 / \sum_1^n |\lambda_i|^2$ (note that $\sum_1^n |\lambda_i|^2 = \||\psi\rangle\|^2$ where $\|\cdot\|$ is the vector norm). General events are subspaces of the Hilbert space.

A matrix can be defined as a *unitary operator* if and only if $UU^\dagger = I = U^\dagger U$, where $\dagger$ indicates the Hermitian conjugate. In quantum probability theory unitary operators can be used to evolve a quantum system or to change the state/space basis: $|\psi'\rangle = U|\psi\rangle$.

Quantum probability theory (see (Vedral, 2007) for a complete introduction) extends standard kolmogorovian probability theory and it is in principle adaptable to any discipline.

2.2 Kernel Quantum Classifier

(Liu *et al.*, 2013) presented a quantum classifier based on the early work of (Chen, 2002). Given an Hilbert space of dimension $n = n_i + n_o$, where n_i is the number of input features and n_o is the number of output classes, they use a unitary operator U to project the input state contained in the subspace spanned by the first n_i basis vectors into an output state contained in the subspace spanned by the last n_o basis vectors: $|\psi^o\rangle = U\left|\psi^i\right\rangle$. Input, $\left|\psi^i\right\rangle$, and output, $|\psi^o\rangle$, states are real vectors, the former having only the first n_i components different from 0 (assigned to the problem input features of every instance) and the latter only the last n_o components. From $|\psi^o\rangle$ they compute the probability of each class as
$P(c_j) = |\psi^o_{ni+j}|^2 / \sum_1^{no} |\psi^o_{ni+i}|^2$ for $j = 1..n_o$.

The unitary operator U for performing instances classification can be obtained by minimising the loss function

$$err(T) = 1/\sum_{j=1}^{|T|} \langle\psi^o_j|\psi^t_j\rangle\,,$$

where T is the training set and $\left|\psi^t\right\rangle$ is the target vector for output probabilities (all zeros except 1 for the target class) for every instance k, using standard optimisation techniques such as Conjugate Gradient (Hestenes, Stiefel, 1952), L-BFGS (Liu, Nocedal, 1989) or ASA (Ingber, 1989).

This classifier exhibits interesting properties managing a classical non-linear problem, the XOR problem, but the simplicity and the low power of this classifier emerge quite clearly when we test it on difficult, though linearly separable, classification problems or on non-linear problems. The classifier is not always able to properly divide the input space into different regions corresponding to the required classes. Moreover, all the decision boundaries have to cross the origin of the feature space, a very limiting constraint for general classification problems, and problems that require strict non-linear decision boundaries cannot be successfully handled by this classifier.

A widely used technique to transform a linear classifier into a non-linear one involves the use of the "kernel trick". A non-linearly separable problem in the input space can be mapped to a higher-dimensional space where the decision borders between classes might be linear. We can do that through the mapping function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$, with $m > n$, that maps an input state vector $\left|\psi^i\right\rangle$ to a new space. The interesting thing is that in the new space, for some particular mappings, the inner product can be calculated by using *kernel* functions $k(x, y) = \langle\phi(x), \phi(y)\rangle$ without explicitly computing the mapping ϕ of the two original vectors.

We can express the unitary operator performing the classification process as a combination of the training input vectors in the new features space

$$\begin{aligned} |\psi^o\rangle &= U\ \ |\phi(\psi^i)\rangle \\ |\psi^o\rangle &= \textstyle\sum_{j=1}^{|T|} |\alpha_j\rangle\ \langle\phi(\psi^i_j)|\ \ |\phi(\psi^i)\rangle \\ |\psi^o\rangle &= \textstyle\sum_{j=1}^{|T|} |\alpha_j\rangle\ \langle\phi(\psi^i_j)|\phi(\psi^i)\rangle \end{aligned}$$

that can be rewritten using the kernel and adding a bias term $|\alpha_0\rangle$as:

$$|\psi^o\rangle = |\alpha_0\rangle + \sum_{j=1}^{|T|} |\alpha_j\rangle\, k(\psi^i_j, \psi^i) \qquad (1)$$

In this new formulation we have to obtain all the $|\alpha_j\rangle$ vectors, $j = 0, ..., |T|$, through an optimisation process similar to the one of the previous case, minimising a standard euclidean loss function

$$\begin{aligned} err(T) = \sum_{j=1}^{|T|} \sum_{k=1}^{no} \left(P_j(c_k) - \psi^t_{j(ni+k)}\right)^2 \\ +\gamma \sum_{j=0}^{|T|} \||\alpha_j\rangle\|. \end{aligned}$$

using a numerical optimisation algorithm, L-BFGS in our experiments, where $P(c)$ is the class probability defined above and $\gamma \sum \||\alpha_j\rangle\|$ is an L_2 regularisation term on model parameters (the real and imaginary parts of $|\alpha_j\rangle$ components).

Once learned a good model from the training set T, represented by the $|\alpha_j\rangle$ vectors, we can use equation (1) and the definition of class probability for classifying new instance vectors.

It is worth noting that the KQC proposed here involves a large number of variables during the optimisation process (namely, $2 * no * (|T| + 1)$) that depends linearly on the number of instances in the training set T. In order to build a classifier applicable to real problems, we have to introduce special techniques to efficiently compute the gradient needed by optimisation methods. We relied on Automatic Differentiation (Griewank, Walther, 2008), avoiding any gradient approximation using finite differences that would require a very large number of error function evaluations. Using such

	Automatic System					
Gold Std.	ang	dis	fea	joy	sad	sur
ang	12	9	1	0	1	7
dis	0	11	3	0	5	2
fea	2	4	5	3	15	1
joy	9	8	1	5	1	6
sad	0	2	0	1	26	1
sur	2	1	1	1	19	6

Table 1: Confusion matrix between the gold standard and the KQC.

	Automatic System					
Gold Std.	ang	dis	fea	joy	sad	sur
ang	16	1	1	2	2	8
dis	6	8	7	0	5	4
fea	3	0	6	4	15	2
joy	10	6	4	7	0	3
sad	0	3	1	1	24	1
sur	2	2	1	1	19	5

Table 2: Confusion matrix between the gold standard and the SVM multiclass classifier proposed in (Joachims *et al.*, 2009).

techniques the training times of KQC are comparable to those of other machine learning methods.

Please, see (Tamburini, in press) for a complete presentation and evaluation of this system.

3 EVALITA 2014 ERT results

We applied the KQC to the EVALITA 2014 Emotion Recognition Task without adapting the system in any way and without devising any specific technique for emotion detection. We participated only at the "closed database" subtask that is devoted to evaluate how much information can be extracted from material coming from a single, professional source of information whose explicit task is to portray emotions and obtain models capable of generalizing to unseen subjects.

As we said in the introduction, we applied a "brute force" approach to this problem: we extracted 1582 features from each utterance using the OpenSMILE package (Eyben *et al.*, 2013) and the configuration file contained in the package for extracting the InterSpeech 2010 Paralinguistic Challenge feature set (Schuller *et al.*, 2010).

In this case $ni = 1582$ and $no = 6$; we excluded from the process all the utterances belonging to the "neutral" class following the task guidelines indications. After a training session using all the utterances and classifications in the Development Set provided by the organisation, we tested the trained classifier on the Test Set executing ten different runs. The outputs of the ten classification processes were mixed and the final results submitted for the evaluation contained the most frequent class chosen by the ten runs for each utterance contained in the Test Set.

The official results assigned the first place to this classifier with a classification accuracy of 36.11%. Table 1 outline the confusion matrix between classes.

We performed some other experiments using a different classifier: the standard Support Vector Machine (SVM) multiclass classifier proposed in (Joachims *et al.*, 2009). This widely diffused state-of-the-art classifier exhibit more or less the same performances of the KQC: 36.67% of accuracy in classifying the six emotions considered in the EVALITA 2014 ERT challenge (the best results are obtained by using a linear kernel and $C = 30$). Table 2 shows the confusion matrix for the SVM multiclass classifier.

4 Discussion and Conclusions

Even if a 36.11% of accuracy allowed this system to be the most accurate in the evaluation campaign (out of two participants), such accuracy is very low; it is much better than the random baseline (16.67%), but certainly not enough for real classification problems. Some emotions, anger, disgust and sadness, can be detected with better reliability, but the other emotions, namely fear, joy and surprise, present classification results very unsatisfactory. The experiments conducted with a different but state-of-the-art classifier, namely a SVM multiclass classifier, present more or less the same picture.

The research question posed in the guidelines "to establish how much information can be extracted from material coming from a single, professional source of information whose explicit task is to portray emotions and obtain models capable of generalizing to unseen subjects" cannot be answered, in our opinion, positively. Emotional recordings taken from a single, even professional, speaker, do not seem to provide enough information to generalise the emotion recognition to other speakers.

Despite the design of KQC is a work in progress

and the it is not free from problems, it exhibits good classification performances, very similar to a state-of-the-art multiclass classifier.

References

Chen J.C.H. 2002. *Quantum Computation and Natural language Processing*. PhD thesis, University of Hamburg.

Eyben F., Weninger F., Gross F. and Schuller B. 2013. Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor. *ACM Multimedia (MM)*, Barcelona, 835–838.

Griewank A. and Walther A. 2008. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Other Titles in Applied Mathematics 105 (2nd ed.), SIAM.

Hestenes M.R. and Stiefel E. 1952. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49 (6), 409–436.

Ingber L. 1989. Very fast simulated re-annealing. *Mathl. Comput. Modelling*, 12 (8): 967–973.

Joachims T., Finley T. and Yu C-N. 2009. Cutting-Plane Training of Structural SVMs. *Machine Learning Journal*, 77 (1): 27–59.

Liu D.C. and Nocedal J. 1989. On the Limited Memory Method for Large Scale Optimization. *Mathematical Programming B*, 45 (3): 503–528.

Liu D., Yang X and Jiang M. 2013. A Novel Text Classifier Based on Quantum Computation. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, 484–488.

Schuller B., Steidl S., Batliner A. 2009. The INTERSPEECH 2009 Emotion Challenge. *Proceedings of Interspeech 2009*, Brighton, 312–315.

Schuller B., Steidl S., Batliner A., Burkhardt F., Devillers L., Muller C. and Narayanan S. 2010. The INTERSPEECH 2010 Paralinguistic Challenge. *Proceedings of Interspeech 2010*, Makuhari, Japan, 2794–2797.

Tamburini F. in press. Are Quantum Classifiers Promising? *Proceedings of The first Italian Computational Linguistics Conference - CLiC*, Pisa.

Vedral V. 2007. *Introduction to Quantum Information Science*. Oxford University Press, USA.

Yeang C.H. 2010. A probabilistic graphical model of quantum systems. *Proceedings, the 9th International Conference on Machine Learning and Applications (ICMLA)*, Washington DC, 155–162.

10.12871/clicit2014223

Forced Alignment on Children Speech

Piero Cosi
Vincenzo Galatà
ISTC-CNR, UOS Padova
Via Martiri della libertà, 2
35137 Padova, Italy
piero.cosi@pd.istc.cnr.it
vincenzo.galata@pd.istc.cnr.it

Francesco Cutugno
Antonio Origlia
Dip.Sc.Fisiche Sez.Informatica,
Università di Napoli "Federico II",
Via Cinthia, I-80126 Napoli, Italy
cutugno@unina.it
antonio.origlia@unina.it

Abstract

English. In this Forced Alignment on Children Speech (FACS) task, systems are required to align audio sequences of children read spoken sentences to the provided relative transcriptions, and the task has to be considered speaker independent.

Italiano. *In questo task di EVALITA 2014 dal nome "Forced Alignment on Children Speech" (FACS), tradotto in "Allineamento Forzato su Parlato Infantile", ai partecipanti è stato richiesto di allineare alcune sequenze audio di parlato letto infantile alle corrispondenti trascrizioni fonetiche. I sistemi in esame sono da considerarsi indipendenti dal parlatore.*

1 Introduction

As with other international evaluation campaigns, guidelines describing the FACS task were distributed among the participants, who were also provided with training data and had the chance to test their systems with the evaluation metrics and procedures used in the formal evaluation. As for FACS, two subtasks were defined, and applicants could choose to participate in any of them:

- phone segmentation
- word segmentation

Two modalities were allowed:

- **closed**: only distributed data are allowed for training and tuning the system
- **open**: the participant can use any type of data for system training, declaring and describing the proposed setup in the final report.

The final formal evaluation is based on Unit Boundary Positioning Accuracy. The evaluation methodology follows the standard described in the documentation of the NIST SCLite evaluation tool (NIST, 2015). The SCLite tool itself was used as scorer.
Finally, there was only one participant for the FACS task and this was the SPPAS system by Brigitte Bigi (Bigi, 2012).

2 Data

Training and development data were available quite in advance of test data and participant had only one week to submit their system results to organizers.

2.1 Training data (adult speech)

About 15 map task dialogues recorded by couples of speakers exhibiting a wide variety of Italian variants from the CLIPS corpus (Savi, Cutugno, 2009). Dialogues length ranges from 7/8 minutes to 15/20 minutes. It is up to participants to split these data in train and development subsets. For each dialogue, the following files are provided:

- full dialogue manually performed transcriptions;
- single turn audio files: PCM-encoded mono WAV files (16KHz). Each file is referenced to turns into the full transcription by means of its name;
- single turn phonetic labeling;
- single turn word labeling.

2.2 Training data (children speech)

About 40 sentences read by 20 female and 20 male children speakers taken from the new CHILDIT-2 corpus (Cosi et al., 2015a) collected by ISTC CNR within the ALIZ-E Project (Cosi

et al., 2015b). Sentences length ranges from 2/3 seconds to 5/6 seconds. It is up to participants to split these data in train and development subsets. For each sentence, the following files are provided:

- full sentences automatic performed transcriptions;
- audio files: PCM-encoded mono WAV files (16KHz). Each file is referenced to turns into the full transcription by means of its name;
- phonetic labeling;
- word labeling,

2.3 Test data (children speech)

About 20 sentences read by 5 unseen new female and 5 unseen new male children speakers from the same CHILDIT-2 training corpus cited above. Sentences length ranges from 2/3 seconds to 5/6 seconds. For each sentence, the following files are provided:

- full sentences automatic performed transcriptions;
- audio files: PCM-encoded mono WAV files (16KHz). Each file is referenced to turns into the full transcription by means of its name.

2.4 Reference data (children speech)

Reference transcriptions were automatically created by a recent KALDI ASR system trained on the FBK CHILDIT corpus. The performances of this system are up to now the best obtained so far on this type of material (Cosi et al., 2015b).

3 Test and Results

As previously stated, unaligned phonetic transcription for each file was provided together with the corresponding wav waveform. The reference phonetic transcription we used for the final evaluation did not contain phones that were not actually pronounced. For the evaluation, we used the SCLite tool from the NIST SCTK toolset (NIST, 2015). Participants were requested to send back to the organizers the results of the alignment process in the same format that was used in the training set. Transcriptions were then converted in the CTM format used to perform evaluation by the SCLITE tool. This was to ensure that the conversion from samples to time instants for the boundary markers would have been performed on the same machine for all the participants and for the reference transcription.

The BNF of the CTM format is defined as follows:

CTM :==< F >< C >< BT >< DUR > phoneme

where :

< F >: the waveform filename;
< C >: the waveform channel;
< BT >: the begin time (seconds) of the phoneme, measured from the start of the file;
< DUR >: the duration (seconds) of the phoneme.

Among the transcription rules, it is relevant to note that the same symbol was used for geminates and short consonants. Only 5 vowels were considered, thus eliminating the difference of open and closed feature. A single allophone was considered bot for nasal phoneme m and n.
The SCLite tool was used to perform the time-mediated alignment (TMA) between the reference and hypothesis files and the phoneme-to-phoneme distance was replaced by the following formulas:

D(correct) = |T1(ref)-T1(hyp)| + |T2(ref)-T2(hyp)|
D(insertion) = T2(hyp)-T1(hyp)
D(deletion) = T2(ref)-T1(ref)
D(substit.) = |T1(ref)-T1(hyp)| + |T2(ref)-T2(hyp)| + 0.001

In this mode, the weights of the phoneme-to-phoneme distances are calculated during the alignment based on the markers distance instead of being preset. Results obtained by the only system participating to FACS on the phone alignment task are presented in Table 1 for three different conditions. The ”Closed A” model was trained using CHILDIT-2 and CLIPS corpora, the ”Closed B” model using only CHILDIT-2 and the ”Open” model using both CHILDIT-2 and CLIPS corpora plus a free corpus available on the web named ”read-Torino”, available at http://sldr.org/ortolang-000894.

	Corr	Sub	Del	Ins	Err	S Err
open	96.7	1.2	2.1	1.1	4.4	48.6
closedA	96.8	1.1	2.1	1.1	4.3	49.8
closedB	96.9	1.2	2.0	1.0	4.1	48.6

Table 1. SCLite Time Mediated Alignment results for the open, closedA, amd closedB case.

Results in Table 2 refer instead to the % of markers correctly assigned within 5, 10, 15, 20, 25 ms.

	5ms	10ms	15ms	20ms	25ms
open	43.5	58.7	75.7	85.5	90.3
closedA	45.2	60.6	77.1	86.7	91.1
closedB	43.7	59.2	76.3	85.9	90.6

Table 2. Percentage of markers correctly assigned within 5,10,15,20,25 ms for the open, closedA, amd closedB case.

4 Conclusion

The main aim of this task was to investigate force alignment techniques on read children speech. We explicitly avoid using spontaneous speech in order to evaluate the force alignment of only children speech quality, without considering the difficulties of having to tackle the problem of elisions, insertions, non-verbal sounds, uncertain category assignments, false starts, repetitions, filled and empty pauses and all similar phenomena typically encountered in spontaneous speech. The SPPAAS systems obtained reasonable high performances in all three presented conditions, and results are quite comparable to the state of the art in other languages. Due to the read speech material, reducing the phone inventory to the target one resulted in no difficulties in the alignment task and, even if it is not statistically significant, a dedicated system (closedB case) resulted the best in term of TMA SCLITE alignment errors.

Unfortunately, the SPPAAS system was the only one participating to the FACS task, thus an incomplete analysis of FACS on children speech had been possible because of the lack of comparison of different systems and techniques.

References

NIST (2015), NIST Scoring Toolkit Version 0.1, ftp://jagar.ncsl.nist.gov/current_docs/sctk/doc/sctk.htm

Brigitte Bigi. 2012. SPPAS: a tool for the phonetic segmentations of Speech. In: *Proceedings of LREC 2012, the eight international conference on Language Resources and Evaluation*, Istanbul (Turkey), 1748-1755, ISBN 978-2-9517408-7-7.

Renata Savy, Francesco Cutugno. 2009. CLIPS: Diatopic, Diamesic and Diaphasic Variations of Spoken Italian. In: *Proceedings of Corpus Linguistics Conference 2009*, http://ucrel.lancs.ac.uk/publications/cl2009/213_FullPaper.doc

Piero Cosi, Giulio Paci, Giacomo Sommavilla, Fabio Tesser. 2015a. Building Resources for Verbal Interaction Production and Comprehension within the Project ALIZ-E. In: *Proceedings of AISV 2015* (to be published - 2015).

Piero Cosi, Giulio Paci, Giacomo Sommavilla, Fabio Tesser. 2015. KALDI: Yet Another AST Toolkit? Experiments on Adult and Children Italian Speech. In: *Proceedings of AISV 2015* (to be published - 2015).

10.12871/clicit2014224

The SPPAS participation to Evalita 2014

Brigitte Bigi
Laboratoire Parole et Langage, CNRS, Aix-Marseille Université,
5 avenue Pasteur, BP80975, 13604 Aix-en-Provence France
brigitte.bigi@lpl-aix.fr

Abstract

English. SPPAS is a tool to automatically produce annotations which includes utterance, word, syllabic and phonemic segmentation from a recorded speech sound and its transcription. This paper describes the participation of SPPAS in evaluations related to the "Forced Alignment on Children Speech" task of Evalita 2014. SPPAS is a "user-friendly" software mainly dedicated to Linguists and open source.

Italiano. *SPPAS è uno strumento in grado di produrre automaticamente annotazioni a livello di parola, sillaba e fonema a partire da una forma d'onda e dalla sua corrispondente trascrizione ortografica. Questo articolo descrive la partecipazione di SPPAS nelle valutazioni relative al task Forced Alignment on Children Speech (allineamento forzato su parlato infantile) di Evalita 2014. SPPAS è un software "open source", è molto semplice da utilizzare ed è particolarmente indicato all'uso da parte di linguisti.*

1 Introduction

EVALITA is an initiative devoted to the evaluation of Natural Language Processing and Speech tools for Italian[1]. In Evalita 2011 the "Forced Alignment on Spontaneous Speech" task was added. Then, in 2014, this task is evolving to "Forced Alignment on Children Speech" (FACS). Nevertheless, as in 2011, systems were required to align a set of audio sequences to the provided relative transcriptions. Forced-aligment (also called phonetic segmentation) is the process of aligning speech with its corresponding transcription at the phone level. The alignment problem consists in a time-matching between a given speech unit along with a phonetic representation of the unit. The goal is to generate an alignment between the speech signal and its phonetic representation. Speech alignment requires an acoustic model in order to align speech. An acoustic model is a file that contains statistical representations of each of the distinct sounds of one language. Each phoneme is represented by one of these statistical representations.

After Evalita 2011 (Bigi, 2012), this paper presents the SPPAS participation to the FACS task. The training procedure and the corpus we used during the development phase to provide a new acoustic model are described.

2 Acoustic models: Training procedure

Phoneme alignment is the task of proper positioning of a sequence of phonemes in relation to a corresponding continuous speech signal. In the alignment problem, we are given a speech utterance along with a given phonetic representation of the utterance. Our goal is to generate an alignment between the speech signal and the phonetic representation.

SPPAS (Bigi, 2011) is based on the Julius Speech Recognition Engine (Nagoya Institute of Technology, 2010). Julius was designed for dictation applications, and the Julius distribution only includes Japanese acoustic models. However since it can use acoustic models trained using the Hidden Markov Toolkik (HTK) (Young and Young, 1994), it can also be used in any other language.

Acoustic models were then trained with HTK using the training corpus of speech, previously segmented in utterances, phonetized and automatically time-aligned. The trained models are Hidden Markov models (HMMs). Typically, the HMM states are modeled by Gaussian mixture densities whose parameters are estimated using an expecta-

[1] http://www.evalita.it/

tion maximization procedure. The outcome of this training procedure is dependent on the availability of accurately annotated data and on good initialization. Acoustic models were trained from 16 bits, 16000 hz wav files. The Mel-frequency cepstrum coefficients (MFCC) along with their first and second derivatives were extracted from the speech in the standard way (MFCC_D_N_Z_0).

The training procedure is based on the VoxForge tutorial[2], except that which from VoxForge uses word transcription as input. Instead, we took as input the proposed phonetized transcription, with or without using the phonetic time-alignment. This procedure is based on 3 main steps: 1/ data preparation, 2/ monophones generation then 3/ triphones generation.

Step 1 is the data preparation. It establishes the list of phonemes, plus fillers, silence and short pauses. It converts the input data into the HTK-specific data format (MLF files). It codes the audio data, also called "parameterizing the raw speech waveforms into sequences of feature vectors" (i.e. convert from wav to MFCC format), using "HCopy" command.

Step 2 is the monophones generation. In order to create a HMM definition, it is first necessary to produce a prototype definition. The function of a prototype definition is to describe the form and topology of the HMM, the actual numbers used in the definition are not important. Having set up an appropriate prototype, a HMM can be initialized by both methods:

- create a flat start monophones model, a prototype trained from phonetized data, and copied for each phoneme (using "HCompV" command). It reads in a prototype HMM definition and some training data and outputs a new definition in which every mean and covariance is equal to the global speech mean and covariance.

- create a prototype for each phoneme using time-aligned data (using "Hinit" command). Firstly, the Viterbi algorithm is used to find the most likely state sequence corresponding to each training example, then the HMM parameters are estimated. As a side-effect of finding the Viterbi state alignment, the log likelihood of the training data can be computed. Hence, the whole estimation process can be repeated until no further increase in likelihood is obtained.

In our script, we train the flat start model and we fall back on this model for each phoneme that fails to be trained with Hinit (if there are not enough occurrences). This first model is re-estimated using the MFCC files to create a new model, using "HERest". Then, it fixes the "sp" model from the "sil" model by extracting only 3 states of the initial 5-states model. Finally, this monophone model is re-estimated using the MFCC files and the phonetized data.

Step 3 creates tied-state triphones from monophones and from some language specificities defined by means of a configuration file. This file summarizes Italian phonemic information as for example the list of vowels, liquids, fricatives, nasals or stop. We created manually this resource, and distribute it on-demand.

3 Corpus description

The training set is made of children recorded while reading some text and is available in the form of time-aligned sentences (one file per sentence). The result of an automatic word segmentation and phoneme segmentation is also available. In addition to the Child corpus, the data of Evalita 2011 were also distributed. Some other data were also collected in the scope of this study: a/ 5300 isolated pluri-syllabic tokens of Italian children, with various recording conditions (often with a poor audio quality); b/ read speech of 41 speakers, recorded at Torino (all speakers are reading the same text), the total duration is 31275.8 seconds. This corpus is available at: http://sldr.org/ortolang-000894

In order to create a development set, some files were randomly picked up of the Child set and manually time-aligned by the author (not phonetician), using Praat with the help of the spectrogram. Then 134 files were annotated, with a duration of 888.77 seconds. It is to be noticed that the phonetization was not changed, only the time-alignments were modified. The time spent to correct the automatic alignments was about 9-10 hours. This development corpus contains 196 silences, 60 fillers, 326 /a/, 218 /e/, 218 /o/ and 192 /i/. For this corpus, 2529 boundaries have to be fixed by the system.

In the evaluations, we propose detailed alignment performances depending on the delta range

[2]http://www.voxforge.org

between the automatic and the reference alignments, using the time-localization of the end-bound of each phoneme.

4 Experiment 1: time-aligned data is good data?

In this experiment, we try to fix which amount of data is required for the initial model of step 2. Only the Child corpus is used: the phonetization of the whole corpus is used in all other stages of the training procedure, and time-aligned data are used only to train the initial model. Results are reported in Figure 1. We can observe that, for this stage of the training procedure, 30 seconds of automatic time-aligned speech are the strict minimum that must be used. It seems that 5 minutes are a good compromise. Then, the data used for this initial model are now fixed (they will not be changed in further experiments): the speech duration for the initial model is 302.72 seconds.

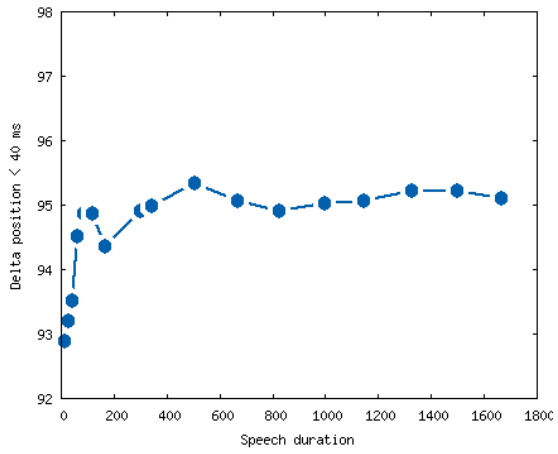


Figure 1: Experiment 1. Results depending of the amount of speech data to train the initial model.

5 Experiment 2: more data is good data?

By fixing the initial model as mentioned in the previous section, we will now evaluate the results while changing the amount of phonetized data (still in step 2, to train the monophones). In this experiment, only the Child corpus is used too. Results are reported in Figure 2. We can observe that from 3 to 10 minutes of data, the differences are very slights, withal we can conclude that more data is good data. However, the differences are not significant for experiments with more than 10 minutes of phonetized speech.

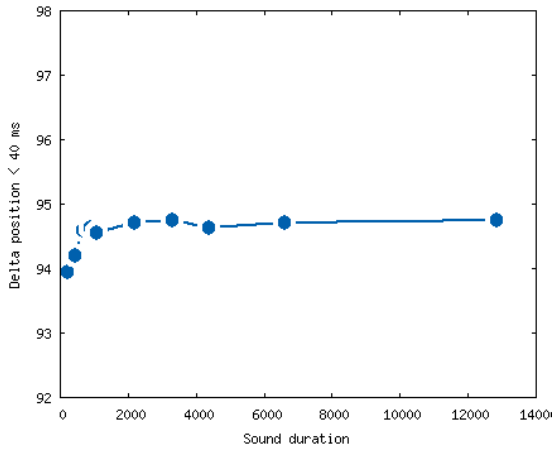


Figure 2: Experiment 2. Results depending of the amount of phonetized speech data.

6 Experiment 3: other data is good data?

We added the data from the CLIPS, distributed by the organizers and then our own data.

Results are reported in Table 1.

Our conclusion is that more data is not good data, and we decided the following: a/ to remove our children corpus of the training data set; b/ to use triphones; c/ to add 5 minutes of time-aligned data of the CLIPS corpus to train the initial model.

7 Final models

We finally trained 3 models by choosing data sets on the basis of the experiments described in the previous sections. The ”Closed A” model was trained using Child and CLIPS corpora, the ”Closed B” model using only Child and the ”Open” model using both Child and CLIPS corpora plus a free corpus available on the web (previously named ”read-Torino”). Results on the development corpus, within a delta of 40 ms, are:

- ”Closed A” 2400 (94.90%)
- ”Closed B” 2406 (95.14%)
- ”Open” 2389 (94.46%)

Figure 3 show detailed results on vowels of the ”Open” model, distributed in SPPAS-1.6.1.

8 Conclusion

During this evaluation campaign, we asked 3 questions and answered within the FACS context. We asked if ”time-aligned data is good data?” and

Model Phonetized Corpus	Monophones # Corr	 %Corr	Triphones # Corr	 %Corr
Only Child	2396	94.74	2404	95.06
Child + dialog-CLIPS	2390	94.50	2395	94.70
Child + read-Torino	2394	94.66		
Child + read-children	2381	94.15		
Child + dialog-CLIPS + read-Torino	2390	94.50	2389	94.46
Child + dialog-CLIPS + read-Torino + read-children	2380	94.11	2362	93.40

Table 1: Results of experiment 3, in a delta less than 40ms.

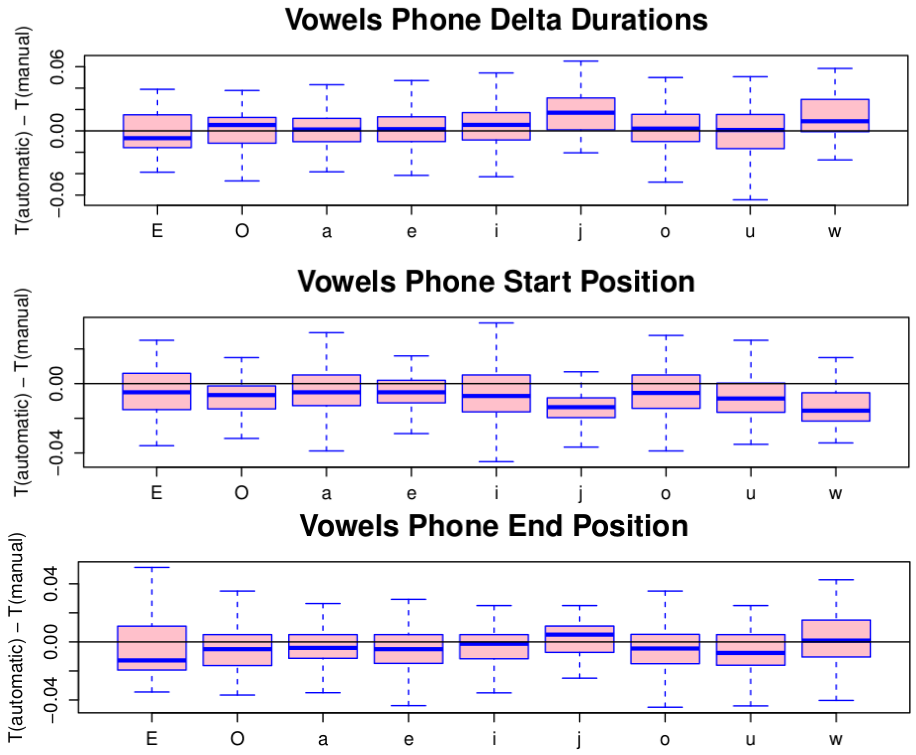


Figure 3: Results on vowels of the ”Open” model.

found that 5 minutes are a good amount of time-aligned data to train the initial model. We asked if ”more data is good data?” and found that at least 10 minutes of phonetized data are required (with more data, the benefits are very slights). We finally asked if ”other data is good data?” and found that the answer is no, a dedicated system is better than a general one (which is not surprisingly).

Acknowledgments

This work has been carried out thanks to the support of the A*MIDEX project (ANR-11-IDEX-0001-02) funded by the ”Investissements d’Avenir” French Government program, managed by the French National Research Agency (ANR). URL of the project:

`http://variamu.hypotheses.org`

References

[Bigi2011] B. Bigi. 2011. SPPAS - Automatic Annotation of Speech, http://www.lpl-aix.fr/∼bigi/sppas/.

[Bigi2012] B. Bigi. 2012. The sppas participation to evalita 2011. *Working Notes of EVALITA 2011*.

[Nagoya Institute of Technology2010] Nagoya Institute of Technology. 2010. Open-source large vocabulary csr engine julius, rev. 4.1.5.

[Young and Young1994] S.J. Young and Sj Young. 1994. The htk hidden markov model toolkit: Design and philosophy. *Entropic Cambridge Research Laboratory, Ltd*, 2:2–44.

10.12871/clicit2014225

Human and Machine Language / Dialect Identification from Natural Speech and Artificial Stimuli: a Pilot Study with Italian Listeners

Antonio Romano **Claudio Russo**
Università degli Studi di Torino, Dip. Lingue e Lett. Str. e Cult. Mod.
Laboratorio di Fonetica Sperimentale "Arturo Genre"
via Sant'Ottavio, 24 I-10124 Torino, Italia
antonio.romano@unito.it clrusso@unito.it

Abstract

English. After a short review of the state of the art, this paper illustrates a selection of the most important Automatic Language Identification and Accent Identification approaches. A series of tasks is presented, providing some evaluation measures about the overall human performance on the basis of language/dialect identification by Italian listeners. Results confirm that humans are able to easily detect linguistic features of languages they have been directly exposed to, thus being able to perform a swift identification when listening even to short samples. Identification rates rise in familiar dialect id. tasks, and a sharp separation is usually established between unknown foreign languages, guessed languages and local varieties of one's own country.

Italian. *Dopo una breve introduzione sullo stato dell'arte, quest'articolo riassume una selezione dei più diffusi approcci all'Identificazione Automatica delle Lingue e degli Accenti (LID/AID). Alcune misure sono offerte riguardo a una serie di test che sono stati svolti per valutare le modalità con cui è avvenuta l'identificazione di una selezione di lingue e dialetti da parte di alcuni uditori italiani. I risultati confermano che gli esseri umani hanno una certa abilità nell'individuare i principali tratti linguistici ai quali sono esposti più spesso e sono, anche per questo, in grado d'identificare agevolmente le lingue conosciute sulla base di campioni di parlato anche piuttosto brevi. Le prestazioni migliorano, infatti, nell'identifica-zione di dialetti con i quali si abbia una certa familiarità. Una separazione netta si può infine stabilire tra lingue straniere sconosciute, lingue indovinate in base a supposizioni e varietà del proprio Paese.*

1 Introduction

Since its origins, the challenge of Automatic Language Identification (*LID*) encountered the problems raised by the presence of dialectal variation and the difficult task of accent identification (*AID*): "the absolute acoustic differences of the native accents is very subtle and sensitive so that they might be an order magnitude smaller than the differences between speech sounds, and be secondary to the individual speaker differences" (Wu *et alii* 2004).

These problems have been tackled by different research teams with a wide set of phone- or acoustic-based techniques (*n-grams*, *phone-lattice* and so on). The state of the art provided by Muthusamy *et alii* (1994) and Geoffrois (2004) during the *MIDL* event of 2004 "Identification des langues et des variétés dialectales par les humains et par les machines" (Paris, France, 29-30 nov. 2004, see Adda Decker *et alii* 2004) needs an update since relevant milestones have been achieved after the *NIST LID* contest of 2003 and the following *NIST LRE* 2005 and 2009. Discriminative *LID* based on *Support Vector Machines* or on *Multi-corpus* and *out-of-set LID* received positive attention since then, and training datasets have been purposefully created and expanded in various *LRE* tasks (following the model of the *Callfriend* corpus, based on labelled speech stuff, and other *LDC* corpora).

Even though the most successful *LID* systems implement more than one component modeling different information types at various levels, several *LID* systems are still nowadays mostly phone-based (cp. Kirchhoff *et alii* 2002, Singer *et alii* 2003, Timoshenko & Bauer 2006; for a review, see, Schultz & Kirchhoff 2006, Wang 2008). Nevertheless, 'acoustic' *LID* systems tend to rely on spectral features in order to extract language-discriminating information encoded within speech productions, whereas language-specific sequences of speech units are traced by 'phonotactic' *LID* systems.

The linguistic information is then usually extracted from the test speech sample with phone recognition modules that rely on either language-

dependent or cross-linguistic acoustic phone models (cp. Yan & Bernard 1995).

According to the scientific literature on human language/dialect identification (Ohala & Gilbert 1981, Romano 1997, Ramus & Mehler 1999), we expect that prosodic level of organisation, such as intonation and rhythm, provides a reliable cue for this purpose (Vaissière & Boula de Mareüil 2004). However, prosodic cues are still less explored in *LID* systems (Navrátil 2006, Leena & Yegnanarayana 2008, Timoshenko 2012) and results of listening tasks aiming to assess the role of the related variables have not yet been achieved for the present study.

After a short review of *LID/AID* models, this paper proposes a discussion about the results of two listening tasks performed by Italian listeners; 54 students were exposed to speech stimuli of 18 foreign languages whereas a selection of 32 of them was asked to identify 20 dialectal varieties.

2 Motivation

Besides the perspective of shedding light on the reasons why automatic speech recognition systems succeed (or fail) when dealing with speech samples encoded in an unknown language, research on human and machine performances in language identification are *per se* interesting.

The challenge for *IT* developers (and for institutions investing on it) is to implement automatic procedures aimed at achieving human performances in language and dialect identification.

On the one hand, that means looking at the inherent language variation in the world (thanks to well documented *DB* and archives, see references) and, on the other hand, trying to emulate human skills in this kind of task.

By the way, also humans do face a challenge when they experience multi-lingual spoken or written communication and are intrigued by language diversity. Whatever their success in dealing with languages which are used in these situations, human beings are amazed by this surprising diversity and are usually challenged to guess the unknown languages they listen to. That explains the large public success of amateur websites such as the "Great language game" (http://greatlanguagegame.com/).

While language variation in specific areas have been captured by various speech/accent archives, significant knowledge about world's languages comes from well-known projects such as *Ethnologue* (Lewis *et alii* 2014) or the *Rosetta* project (rosettaproject.org/). Academic research recently yielded a relevant progress thanks to authoritative sources such as *WALS*, but has also benefited by recent contributions such as *Langscape* or *Phoible*. These projects gathered questionable but useful speech samples as well as phonetic/phonological and bibliographic data on sound structure (this aspect founds a consolidated reference in the *UCLA Phonetic Segment Inventory Database* and the more recent *Lyon-Albuquerque Phonological Systems Database*).

As the individual sensitivity is generally very poor when facing dialectal variation outside the area of origin or residence, so is the knowledge gathered about such variation in large repository sites. Furthermore, dialectal variation is heterogeneous within the different countries. In some areas, a monolingual situation is attested, with potential accent variation throughout the whole territory, but some other regions may be characterised by a jumble of different languages and each of them strongly affected by dialectal variation (cp. Tsai & Chang 2002). This is the situation of Italy and its surrounding countries.

Languages and dialects spoken in Italy are surveyed and discussed in several dialectological studies (among others, Maiden & Parry 1997, Loporcaro 2009) and a remarkable quantity of lexical and phonetic data is provided by linguistic atlases such as the *ALI* (Massobrio *et alii* 1996) who helped in the definition of the dataset (§3.2). Nevertheless, the available information is hardly exploitable for testing since no speech samples are included and data is not intended for IT purposes or language identification tasks. Experiments on the perception of foreign accent in Italian are carried out by some research teams (De Meo *et alii* 2011), but native accented speech is less studied and the general knowledge of Italian speakers about regional varieties/dialects is almost completely ignored.

2.1 Automatic *LID*/*AID* methods

Within the last twenty years, universities from all over the world jointly worked with *IT* companies to produce effective automated speech recognition systems. Thanks to this striking cooperative effort, the research community witnessed a wide range of different techniques, which can be roughly classified as:

- techniques based on parallel phone recognition for phone lattice classification (*PPLRM*; cp. Gauvain *et alii* 2004). These approaches relied mostly on language-dependent *n-gram* models and context-independent phone models to classify the salient features of phonotac-

tic traits. Both context-dependent Hidden Markov Models (*CD-HMM*) and null-grammar *HMM* have been exploited by this particular approach (Damashek 2005, Suo *et alii* 2008);

- techniques focused on spectral change representation (*SCR*) and extraction of prosodic features. These approaches usually look at utterances as collections of independent spectral vectors. For accent identification (*AID*) purposes, such vectors are combined in a supervector that is assigned to each speaker; to achieve *LID*, the vector collection is usually modeled by Gaussian Mixture Models (*GMM*s) or similar (Kirchhoff *et alii* 2002). Within these approaches, an unusual solution has been explored with the Bag-of-sounds (*BOS*) technique, which exploits a universal sound recogniser to create a sound sequence that is converted into a count vector at a second stage. The classifier being trained, the *BOS* technique does not need any acoustic modelling to add new language capabilities;
- hybrid techniques have been refined thanks to different technologies (such as Deep Neural Networks, *DNN*s, used as state probability estimators; Lopez Moreno *et alii* 2014). Recently, further attempts towards *GMM*-free approaches have been made, aiming at improving segmentations through online interaction with a parameter server and graph-based semi-supervised algorithms for speech processing (Liu & Kirchhoff 2013).

3 Tasks for human listeners

Since human perception of identification cues are unconscious, listening experiments are needed in order to empirically assess in which way human language identification occurs.

In this research, three listening tasks have been proposed to test human abilities in language and dialect identification.

Testing scripts and soundwave files were freely distributed at the following website: http://www.lfsag.unito.it/evalita2014/index.html. The execution of the listening tasks required the installation of the *PRAAT* software and the creation of a *HMDI* folder on the PC. Instructions on how to carry out each experiment were illustrated by a *.pps* slideshow.

HMDI (see §3.1 and 3.4) was a task aiming at testing human abilities to identify languages from short speech samples.

The two following tasks *HMDI_DIA* and *HMDI_TON* were intended to test dialect identification by natural and synthetic speech samples. *HMDI_DIA* (see §3.2 and 3.5) was a task mainly intended for listeners living in Italy and it aimed at testing their abilities to identify dialectal varieties whereas *HMDI_TON* was conceived to test the possibility to identify dialect just relying on prosodic values extracted from real sentences. Results of the latter are not reported here.

3.1 First Dataset (*HMDI*)

The *HMDI* task was based on a sample of 18 languages represented by natural stimuli recorded in a soundproof booth. Two samples based on passages from a local version of the IPA narrative "The North Wind and the Sun" were submitted to the listeners' judgment. All the recordings are original and belong to a larger ongoing speech archive available at the *LFSAG*.

All the speakers were women aged between 20 and 28. Stimuli are coded with a number corresponding to each language as it follows:

1. Albanian (Durrësi-Duras accent)
2. Arabic (Tunisian accented *SMA*)
3. Baoulé (from Bouaké, Ivory Coast)
4. Chinese (from the Jiangsu region)
5. Farsi (from Tehran)
6. Bavarian German (Südtyrolian dialect)
7. Hebrew (from Jerusalem)
8. Hungarian (from Eger)
9. I.-Veneto (from Vodnjan-Dignano, Istria)
10. Latvian (from Riga)
11. Macedonian (from Bitola)
12. Polish (from Krakow)
13. Portuguese (Capeverdean accent)
14. Romanian (from Braşov)
15. Serbian (from Beograd)
16. Spanish (from Buenos Aires, Argentina)
17. Sardinian (from Orosei)
18. Vietnamese (Hanoi accent).

Speech samples have a variable length (between 7.2 and 13.3 s) and more or less the same number of syllables belonging to a text which corresponds to the narrative's last passages: "And so the North Wind was obliged to confess that the Sun was the stronger of the two. Did you like the story? Do you want to hear it again?".

Listeners sat before a PC monitor wearing a headset and decided when to run the PRAAT script. Speech stimuli for this experiment were played twice in random order and listeners were asked to select the corresponding language label in an interactive window as quickly as possible.

The overall duration of the each test session was about 6-10 min.

3.2 Second Dataset (*HMDI_DIA*)

The *HMDI_DIA* task relied on a sample of 20 dialects. Even in this case, stimuli were extracted from a local version of "The North Wind and the Sun".

All the speakers were female aged between 20 and 28 except for one who was in her 40s.

The task was intended for Italian listeners and is mainly based on samples selected from dialects which are spoken in Italy or nearby but includes several dialects of foreign languages as distractors/control languages.

The test was administered by means of a PRAAT script (see above) and through an interactive window allowing the listener to choose a language label on the screen after listening to each of the 20 stimuli (randomly played once). Since the task was intended for Italian listeners, languages were labelled in Italian.

The stimuli were taken from recordings collected for the following languages: *Arabo M.* (Moroccan Arabic), *Arabo T.* (Tunisian accented S.M. Arabic), *Napoletano* (Neapolitan), *Occitano P.* (Piedmont Occitan), *Pugliese* (Apulian), *Polacco K.* (Polish from Krakow), *Polacco W.* (Polish from Wrocław), *Piemontese* (Piedmontese from Saluzzo), *Portoghese C.V.* (Capeverdean Portuguese), *Portoghese T.E* (Portuguese from East Timor), *Romeno V.* (Romanian from Braşov), *Romeno M.* (Moldavian from Chişinau), *Siciliano Or.* (East Sicilian from Catania), *Siciliano Occ.* (West Sicilian from Erice), *Siciliano Mer.* (Southern Sicilian from Pachino), *Salentino* (Sallentinian from Mesagne), *Spagnolo A.* (Argentinian Spanish), *Spagnolo V.* (Venezuelan Spanish), *Sardo* (Sardinian), *I.-Veneto* (Veneto-Istrian dialect from Vodnjan-Dignano).

Even in this dataset, the length of the stimuli was well below the usual *LID* values and it was variable between 5.5 and 13.2 s.

3.3 Listeners' samples

Listeners were 54 students, or visiting students at the Uni.TO, aged between 18 and 35 (34 women and 20 men; 93% were students of foreign languages). 37% were first-degree students and the remaining 63% was almost equally represented by MA and PhD students. 17% of the sample was constituted by students of foreign origins (2 Spanish, 2 Romanian, 2 Macedonian, 1 Moroccan, 1 Iranian and 1 Albanian).

For the *HMDI_DIA* task the sample was reduced to 34 listeners (mainly of Italian origins or living since various years in Italy and very proficient in Italian). Many of them had Piedmontese origins (24, that is 71%) and declared a passive knowledge of a local dialect (6 of them of another dialect spoken in Italy: 2 Sicilian, 2 Apulian and 2 Sardinian). Furthermore, 14 listeners (41%) reported an active competence of a foreign language (1 Spanish, 1 Romanian) or another dialect spoken in Italy (3 Calabrian, 3 Sicilian, 3 Apulian, 2 Sallentinian and 1 Sardinian).

3.4 Evaluation measures for *HMDI*

Generally speaking, for the first task (*HMDI*) listeners answered correctly 713 times, which means that 36.7% languages of the tested sample have been correctly identified.

A negligible learning effect has been observed from the first to the second passage of the same stimulus: 350 correct responses were collected for the first repetition vs. 363 for the second one.

Individual responses were displayed in confusion plots such the one showed in Fig. 1, whereas overall results are summarised in Fig. 2.

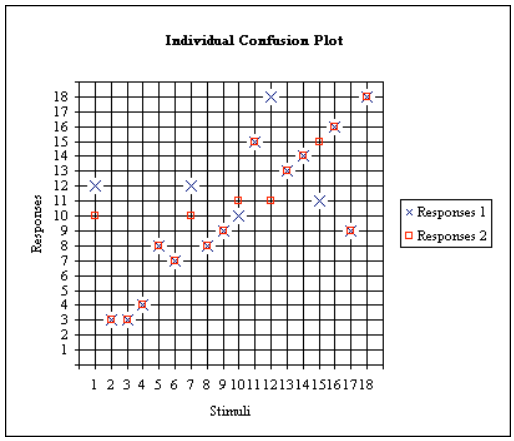


Fig. 1 – Individual plot of responses given to each pair of language stimuli.

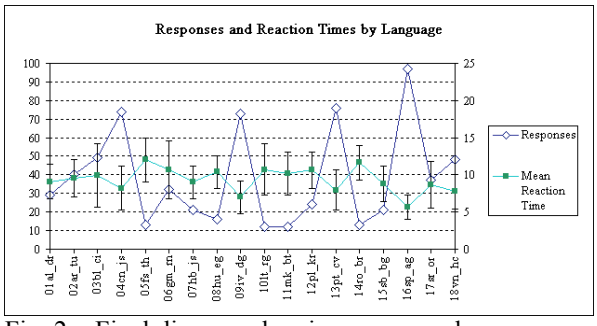


Fig. 2 – Final diagram showing scores and mean reaction times for each test language.

All the responses were statistically analysed by using *R* functions and scripts. Of course, re-

sults have not been assessed in *DET* curves diagrams, as for automatic systems, since only one sample per language was tested. Even though Miss probabilities and False Alarm rates could be extensively discussed for human listener too (cp. Swets 1964), the sample was reduced (and responses were highly non-linear). Therefore, general results (plotted in Fig. 3 and summarized in table I) are discussed in a more adapted way.

As shown in Fig. 3, the listeners responded variously. The top-four, most-identified languages were Spanish (row 16), Portuguese (r. 13), Chinese (r. 4) and Veneto-Istrian (r. 9).

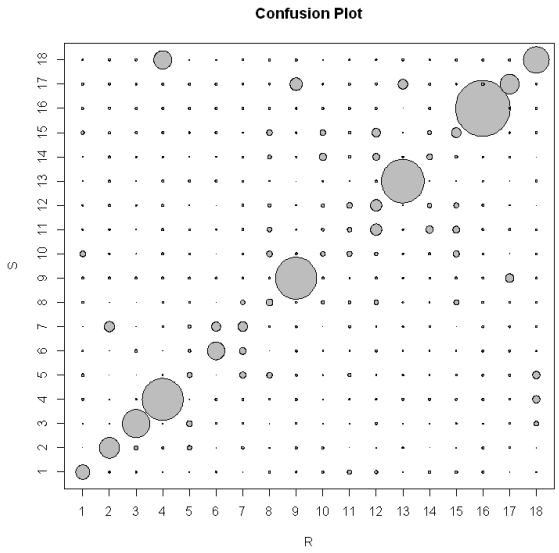


Fig. 3 – Confusion plot for the 18 stimuli (*S* axis) and responses (*R* axis) for the first task. See the text for language codes (§3.1).

The four least-identified languages were Latvian (r. 10), Macedonian (r. 11), Romanian (r. 14) and Farsi (r. 5). The error rate (*ER*) for Spanish, Portuguese, Chinese and Veneto-Istrian is 6%, 26%, 29% and 29% respectively, whereas it rises to 87-89% for the less identified languages. It is worth noticing how Latvian has been uniformly confused among Arabic, Hungarian, Portuguese and Serbian. Macedonian has been confused mostly with Polish, Serbian and Romanian and the latter with Latvian, Polish and Hungarian. Finally, it is interesting to notice how the listeners identified Vietnamese (r. 18) despite their lack of any kind of knowledge about it. A similar score was achieved for Baoulé (r. 3).

When guessing the right answer, the listeners expressed their preference for some languages in particular: Polish, Portuguese and Chinese above others. Conversely, Sardinian, Arabic and Südtyrolian German scored preference values below their actual presence in the task. This may signal a sort of prototypical reference role of the former languages for listeners of this almost homogeneous sample.

Finally, the dispersion plot in Fig. 4 allows establishing an inverse proportionality between the number of correct answers and the reaction times (RT) as a general trend for all the listeners. RT were significantly lower for the declared known languages (5,4 s) than for unknown or guessed languages (10,7 s; a two-sample Welch t-test gave t = -9.36, df = 65.98, p-value = 1.009e-13).

Table I. Confusion matrix (Task HMDI, see §3.1)

	Responses																	
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
01al_dr	29	1	1	2	3	1	3	8	1	8	12	10	2	8	8	0	5	2
02ar_tu	4	40	11	0	11	3	8	2	0	7	3	2	1	4	2	0	1	5
03bl_ci	2	2	49	3	14	1	1	3	1	3	2	0	5	2	1	2	1	12
04cn_js	0	2	1	74	5	0	0	1	0	2	1	0	0	0	0	0	1	17
05fs_th	8	4	4	6	13	4	14	14	0	6	10	2	1	2	1	0	0	15
06gm_rn	1	3	9	3	9	32	16	4	0	2	2	8	2	2	5	0	0	6
07hb_js	2	22	3	1	9	18	21	7	0	4	8	2	1	1	4	0	0	1
08hu_eg	8	3	4	0	7	3	12	16	0	10	8	11	1	2	12	0	1	6
09iv_dg	0	0	0	0	0	0	0	0	73	0	2	0	1	1	0	8	19	0
10lt_rg	14	1	3	0	1	3	1	13	0	12	13	10	7	8	15	1	1	1
11mk_bt	6	1	3	0	1	2	0	12	2	8	12	24	1	15	16	0	0	1
12pl_kr	7	0	1	0	2	4	0	7	2	9	14	24	3	12	13	0	2	4
13pt_cv	2	0	2	0	0	0	1	2	4	1	2	0	76	5	3	1	5	0
14ro_br	5	0	1	1	2	2	6	11	1	17	8	16	7	13	8	1	1	4
15sb_bg	9	0	0	0	1	0	2	14	1	13	8	19	5	11	21	0	0	0
16sp_ag	0	0	0	0	0	0	0	0	1	0	0	0	4	0	1	97	1	0
17sr_or	0	0	1	0	0	0	1	1	23	1	0	0	21	8	1	9	37	1
18vn_hc	1	0	7	35	5	2	1	0	0	2	1	1	0	1	0	0	0	48

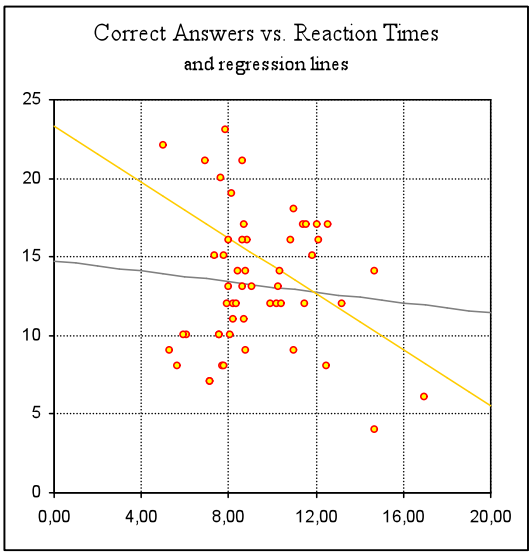


Fig. 4 – Dispersion plot of the number of correct answers vs. Reaction time for all the listeners.

3.5 Evaluation measures for *HMDI_DIA*

As for the second task (*HMDI_DIA*), listeners answered correctly 289 times out of 680 stimuli, which means a 42.5% score of language/dialect identification. Dialects within the Italo-Romance space were correctly identified at 57.3% (184 judgments out of 321).

We did not expect the Italian listeners to identify the dialects of those foreign languages which had not been identified in the first task (see §3.4); these stimuli were intended for foreign listeners and acted as distractors/reference noise for native Italian listeners. Conversely, the possibility of discrimination among Eastern, Western and Southern Sicilian was too ambitious for the current composition of the listener sample and served for comparisons. Partial scores are then collapsed into a total score (01-05 for the foreign languages and 10 for Sicilian, see *Table II*).

Fig. 5 shows the overall sample's responses in the second task. The plot clearly highlights that local dialects are perceived as such, in contrast with foreign languages. Appropriate responses to stimuli in languages other than Italian dialects are classified in the small, top-left square of Table II: while it is true that some listeners failed to positively identify some foreign languages (i.e. Polish and Romanian), they straightforwardly perceived such languages as unrelated to Italian dialects. The bigger, bottom-right square summarises the responses to dialect stimuli: again, the listeners generally identified the language they had listen to, Sardinian being the only exception. Sardinian has been correctly identified 8 times and confused 5 times with Veneto-Istrian, Sicilian and Portuguese, and 4 times with Spanish (minor confusion with other languages and dialects aside), with an extraordinary *ER* of 76%.

It is worth noticing that Sardinian has been perceived as a foreign language in 32% of cases whereas Veneto-Istrian has been confused with a foreign language in only one case (with Spanish).

Foreign languages have been identified as such with a 96% accuracy (325 correct answers), but listeners' also scored a 94% accuracy ratio in recognising dialect data as such. Of course, specific dialects scored 100% from listeners who previously declared a competence of them. Generally speaking, we may say instead that Sicilian (and Neapolitan), as well as Veneto-Istrian, provided good references for southern and northern broad dialectal areas for listeners who were not trained to detect subtler differences.

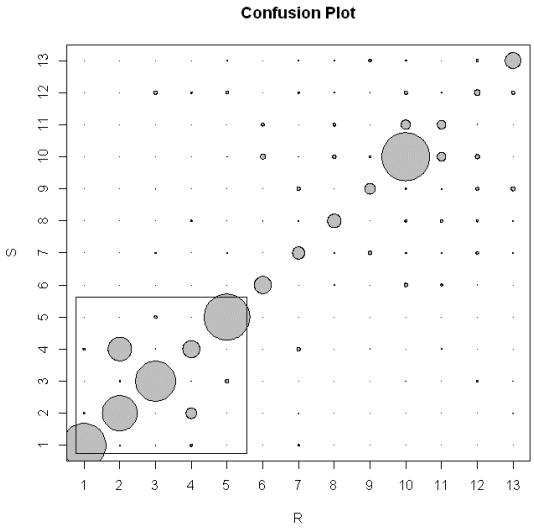


Fig. 5 – Confusion plot for the 13 stimuli (*S* axis) and responses (*R* axis) for the second task. See table II for language codes.

Table II. Confusion matrix (HMDI_DIA, §3.2)

	Responses												
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
01AR	61	1	1	3	0	0	2	0	0	0	0	0	0
02PL	2	49	0	15	0	0	1	0	0	0	0	0	1
03PT	0	2	56	1	5	0	0	0	0	1	0	2	1
04RO	3	33	1	24	0	0	5	0	1	0	1	0	0
05SP	0	0	4	0	64	0	0	0	0	0	0	0	0
06NA	0	0	0	0	0	24	0	1	0	5	3	1	0
07OC	0	0	2	0	1	0	17	1	5	2	1	4	1
08PG	0	0	0	2	0	1	1	19	0	3	4	3	1
09PM	0	0	1	0	0	0	5	0	15	2	1	4	6
10SC	0	0	0	0	1	7	1	5	3	67	12	6	0
11SL	0	0	0	0	0	4	0	4	0	13	12	1	0
12SR	0	0	5	2	4	0	2	1	1	5	1	8	5
13IV	0	0	0	0	1	0	1	1	4	2	0	3	22

4 Task for *LID/AID* systems

The speech samples presented in §2 were also designed for testing machine performances after a training of the *LID/AID* systems of each participant on longer and multispeaker samples downloadable in a *HMDI_TRAINING* folder. Candidates in testing their *LID/AID* systems were also invited to run it on telephonic or noisy samples available in the *HMDI_NOISY* folder.

4.1 Participation-results

Unfortunately, no participant chose to fully complete the proposed task procedure. Only three research teams previously showed their interest in it, but no documentation has been produced.

As a first attempt to compare human performances and the possibilities for automatic procedure to approximate them, we tested a few variables in our data that may prompt a more extensive pilot study on Italian dialects identification.

We particularly took into account listeners' comments pointing out the relevance for them of intonation cues. By the way, some listeners easily distinguished Polish and Portuguese, as well as Sardinian and Apulian, from the other languages or dialects, and reported that they relied on the overwhelming presence of fricative sounds in the stimuli for these varieties.

In facts, the stimuli used for Polish and Portuguese are characterised by the presence of 26 and 16 sharp fricative segments, respectively, vs. e.g. the number of fricatives affecting the passages in other languages (e.g. in the stimuli for Vietnamese, Baoulé or even Spanish and Veneto-Istrian, fricatives were limited to a selection of 6-9 fricatives with generally flat spectrum).

Overall variables accounting for general spectral properties, such as *CoG*, standard deviation (*st.dev*) or spectral tilt, are well taken into account for speech recognition and *LID* purposes (Wu *et alii* 2004). In our case, *CoG* and *st.dev* alone account for the discrimination of the two language groups (*st.dev* ranged over 1000 Hz for the former, whereas it was particularly low, < 700 Hz, for the latter). Even the zero-crossing scores discriminated the two groups, with higher values for 'sharp fricative languages' (> 2000 zc/s) vs. 'flat fricative languages' (< 1300 zc/s). Nevertheless, familiarity as well as areal, lexical or phonotactic features must have played a discriminating role within the same group, so allowing these listeners to distinguish e.g. Portuguese from Polish or Sallentinian from Occitan (all mostly ignored by the listeners). In particular, local prosodic signals and phonotactic regularities (whose importance is highlighted since Arai 1995; cp. Tong *et alii* 2006, 2009) are supposed to provide cues for human dialect identification.

5 Conclusion

Since no report about automatic *LID* on the proposed language/dialect datasets was delivered, this paper aimed at provisionally surveying only the main results of a series of experiments on language/dialect identification carried out with the help of a sample of 54 Italian listeners.

In particular, after a short review of the most widespread techniques in automatic *LID*, a pilot study has been proposed, which explores responses and reaction times and try to match individual scores with linguistic biographies.

An areal sensitivity has been confirmed and a clear-cut separation emerged between known, guessed and unknown dialects in terms of scores and reaction times.

The next step will consist in testing how a training may improve listeners' performances.

6 References

Adda Decker M. *et alii* (eds.) (2004). *Identification des langues et des variétés dialectales par les humains et par les machines - Proc. of MIDL* (Paris, France, Nov. 2004), Paris, ENST.

ALI – Massobrio L. *et alii* (1996-). *Atlante Linguistico Italiano* (http://www.atlantelinguistico.it/, last accessed July 2014).

Arai T. (1995). "Automatic language identification using sequential information of phonemes". *IEICE Trans.*, E78-D/6, 705-711.

Damashek M. (2005). "Gauging Similarity with n-Grams: Language Independent Categorization of Text". *Science*, 267/10, 843-848.

De Meo A., Vitale M., Pettorino M. & Martin Ph. (2011). "Acoustic-perceptual credibility correlates of news reading by native and non-native speakers of Italian". *Proc. of ICPhS2011* (Hong Kong, August 2011), 1366-1369.

Gauvain J.L., Messaoudi A. & Schwenk H. (2004). "Language Recognition Using Phone Lattices". *Proc. of ICSLP '04* (Jeju Island, South Korea, October 2004), 1283–1286.

Geoffrois E. (2004). « Identification automatique des langues : techniques, ressources, et évaluations ». In Adda Decker *et alii* (eds.), 43-44.

Suo H., Li M., Liu T., Lu P. & Yan Y. (2008). "The Design of Backend Classifiers in PPRLM System for Language Identification". *EURASIP Journal on Audio, Speech and Music Processing*, 6 p. (doi: 10.1155/2008/ 674859).

Kirchhoff K., Parandekar S. & Bilmes J. (2002). "Mixed-memory Markov Models for Automatic Language Identification". *Proc. of ICASSP2002* (Orlando, USA, May 2002), 2841-2844.

Ladefoged P. & Maddieson I. (1996). *The sounds of the world's languages*. Oxford, Blackwell.

Langscape – Maryland Language Science Center, University of Maryland - *Language Identification Tool* and *Language Familiarization Game* (http://langscape.umd.edu/, last accessed 27 Oct. 2014).

LDC – Linguistic Data Consortium - University of Pennsylvania (https://www.ldc.upenn.edu/, last accessed 27 Oct. 2014).

Leena M. & Yegnanarayana B. (2008). "Extraction and representation of prosodic features for language and speaker recognition". *Speech Communication*, 50, 782–796.

Lewis M.P., Simons G.F. & Fennig Ch.D. (eds.) (2014). *Ethnologue: Languages of the World*. Dallas, SIL International (17th ed.; http://www.ethnologue.com, last accessed 14 Oct. 2014).

Liu Y. & Kirchhoff K. (2013). "Graph-Based Semi-Supervised Learning for Phone and Segment Classification". *Proc. of Interspeech 2013* (Lyon, France, August 2013), 1839-1842.

Lopez-Moreno I., Gonzalez-Dominguez J., Plchot O. *et alii* (2014). "Automatic Language Identification Using Deep Neural Networks". *Proc. of ICASSP 2014* (Florence, Italy, May 2014), 5374-5378.

Loporcaro M. (2009). *Profilo linguistico dei dialetti italiani*. Roma-Bari, Laterza.

Maiden M. & Parry M. (eds.) (1997). *The Dialects of Italy*. London-New York, Routledge.

Muthusamy Y.K., Barnard E. & Cole R.A. (1994). "Reviewing automatic language identification". *IEEE Signal Processing Magazine*, 11/4,. 33-41.

Navrátil J. (2006). "Automatic Language Identification". In Schultz & Kirchhoff (eds.), 233-268.

Ohala J.J. & Gilbert J.B. (1981). "Listeners' ability to identify languages by their prosody". In: P. Leon & M. Rossi (eds.), *Problèmes de Prosodie: vol. 2*, Paris, Didier, 123-131.

Phoible – Moran S. & McCloy D. & Wright R. (eds.) 2014. *PHOIBLE Online*. Leipzig, Max Planck Institute for Evolutionary Anthropology (http://phoible.org/, last accessed 24 Oct. 2013).

PRAAT – Boersma P. & Weenink D. (1995-2013). *Praat: doing phonetics by computer* (http://www.fon.hum.uva.nl/praat/ v. 5.3.03, 2011).

Ramus F. & Mehler J. (1999). "Language identification with suprasegmental cues: A study based on speech resynthesis". *J. A. S. A.*, 105/1, 512-521.

R-language – The R Project for Statistical Computing (http://www.r-project.org/, last acc. 14 Oct. 2014).

Romano A. (1997). "Persistence of prosodic features between dialectal and standard Italian utterances in six sub-varieties of a region of Southern Italy (Salento): first assessments of the results of a recognition test". *Proc. of EuroSpeech97* (Rhodes, Greece, September 1997), 175-178.

Schultz T. & Kirchhoff K. (eds.) (2006). *Multilingual Speech Processing*. Amsterdam, Elsevier Academic Press.

Singer E., Torres-Carrasquillo P.A., Gleason T.P., Campbell W.M. & Reynolds D.A. (2003). "Acoustic, phonetic, and discriminative approaches to automatic language identification". *Proc. of Eurospeech 2003 - Interspeech 2003* (Geneva, Switzerland, September 2003), 1345-1348.

Swets J.A. (1964). *Signal detection and recognition by human observers: contemporary readings*. New York, Wiley & sons.

Timoshenko E. & Bauer J.G. (2006). "Unsupervised adaptation for acoustic language identification". *Proc. of ICSLP2006* (Pittsburgh, USA, September 2006), 409-412.

Timoshenko E. (2012). "Rhythm Information for Automated Spoken Language Identification". *PhD Thesis*, Technischen Universität München (https://mediatum.ub.tum.de/doc/1063301/106330.pdf, last accessed 28 Oct. 2014).

Tong R., Ma B., Li H. & Chng E.S. (2009). "A target-oriented phonotactic front-end for spoken language recognition". *IEEE Transactions on Audio, Speech and Language Processing*, 17/7, 1335-1347.

Tong R., Ma B., Zhu D., Li H. & Chng E.S. (2006). "Integrating Acoustic, Prosodic and Phonotactic Features for Spoken Language Identification". *Proc. of ICASSP2006* (Toulouse, France, May 2006), 205-208.

Tsai W.H. & Chang W.W. (2002). "Discriminative training of Gaussian mixture bigram models with application to Chinese dialect identification". *Speech Communication*, 36, 317-326.

Vaissière J. & Boula de Mareüil Ph. (2004). "Identifying a language or an accent: from segments to prosody". In Adda Decker *et alii* (eds.), 1-4.

WALS – B. Comrie *et alii* (eds.), *World Atlas of Linguistic Structures* (http://wals.info/, last accessed 14 Dec. 2013).

Wang L. (2008). "Automatic Spoken Language Identification". *PhD Thesis*, The Univ. of New South Wales (http://www.nicta.com.au/pub?doc=1784, last accessed 28 Oct. 2014).

Wu T., Van Compernolle D., Duchateau J., Yang Q. & Martens J.P. (2004). "Spectral Change Representation and Feature Selection for Accent Identification Tasks". In Adda Decker *et alii* (eds.), 57-61.

Yan Y. & Bernard E. (1995). "An approach to automatic language identification based on language-dependent phone recognition". *Proc. ICASSP '95* (Detroit, USA, May 1995), 3511-3514.

10.12871/clicit2014226

SASLODOM: Speech Activity detection and Speaker LOcalization in DOMestic environments

Alessio Brutti, Mirco Ravanelli, Maurizio Omologo
Center for Information and Communication Technology - Fondazione Bruno Kessler
via Sommarive 18, 38123, Trento
{brutti, mravanelli, omologo}@fbk.eu

Abstract

English. This paper describes the design, data and evaluation results of the speech activity detection and speaker localization task in domestic environments (SASLODOM) in the framework of the EVALITA 2014 evaluation campaign. Domestic environments are particularly challenging for distant speech recognition and audio processing in general due to reverberation, the variety of background noises, the presence of interfering sources as well as the propagation of acoustic events across rooms. In this context, a crucial goal of the front-end processing is the detection and localization of speech events generated by users within the various rooms. The SASLODOM task aims at evaluating solutions for both activity detection and source localization on corpora of multi-channel data representing realistic domestic scenes.

Italiano. *In questo articolo viene presentato il database, le metriche e i risultati della valutazione del task SASLDOM all'interno della campagna di valutazione EVALITA 2014. Gli ambienti domestici sono particolarmente sfidanti per le tecnologie di riconoscimento vocale ed elaborazione audio in genere, a causa del riverbero, della varietá di rumore di fondo, della presenza di interferenti e infine a causa della propagazione degli eventi acustico attraverso le stanze. In questo contesto un aspetto cruciale del front-end acustico è la capacità di rilevare e localizzare gli eventi acustici generati dall'utente nelle varie stanze. Il task SASLODOM mira a valutare soluzioni di rilevamento del parlato e localizzazione della sorgente su due database multicanale che rappresentano tipiche scene domestiche.*

1 Introduction

The SASLODOM challenge, within the framework of EVALITA 2014, addresses the problem of the detection in time and localization in space of speech events in domestic contexts. A considerable number of applications could benefit from natural speech interaction with distant microphones (Wölfel and McDonough, 2009). In particular, the possibility to control by voice the devices and appliances of an automated home has recently received a significantly growing interest. This scenario is being targeted by the EU project DIRHA[1] (Distant-speech Interaction for Robust Home Applications) focusing on motor-impaired users, whose life quality can considerably improve thanks to speech-driven automated home.

A desirable property of a distant-speech interaction system in domestic contexts is the capability to be "always-listening" and to always accept commands or requests from the users. This feature represents a noteworthy challenge, as the system must be able to keep as low as possible the rate of false alarms, generated by acoustic events that are not intended to convey any message addressed to the recognition system, while at the same time it must be able to detect any speech command, independently of the current environmental conditions and without introducing constraints on the user position and orientation. Hence, fundamental features of the front-end processing component are a robust Speech Activity Detection (SAD) and Source LOCalization (SLOC). A correct identification of time boundaries, room and spatial coordinates of each speech event is essential for the targeted interactive scenario. In fact, the efficiency of

[1] http://dirha.fbk.eu

a dialogue manager or of a command-and-control system, strongly depends on the performance of the ASR system in the right room: in several cases the system must be able to serve the user also on the basis of the location where the speech command has been given (i.e., the command "open the window" implies that the window to open is located in the same room.). The critical role of the SAD component both in distant-talking ASR and in acoustic event classification has been studied in (Macho et al., 2005).

There is a wide literature addressing SAD techniques. Early works on specific speech/non-speech segmentation focused on close talking interaction and were based on the use of energy thresholding and zero-crossing features (Junqua et al., 1994), in some cases exploring the use of noise reduction (Bouquin-Jeannes and Faucon, 1995). Also, well-known features among the speech recognition community, like MFCCs and PLP, have been used for audio event detection (Portelo et al., 2008; Trancoso et al., 2009). Additionally, techniques based on Spectral Variation Functions (SVF) (DeMori, 1998) or other spectro-temporal features (Pham et al., 2008) can be exploited to discriminate speech from stationary background noise, even under unfavorable SNR conditions. Various machine learning methods (Shin et al., 2010), are used to provide a final classification of the audio events such as Gaussian Mixture Models (GMMs) (Chu et al., 2004), Support Vector Machines (SVMs) (Guo and Li, 2003), Hidden Markov Models (HMMs) and Bayesian Networks (Cai et al., 2006). Recently, solutions relying on Deep Neural Networks (DNN) have been employed (Zhang and Wu, 2013). Finally, the availability of multiple acquisition channels permits the implementation of multi-channel processing (Wrigley et al., 2005; Dines et al., 2006), or the adoption of different feature sets, eventually based on the spatial coherence at two or more microphones (Armani et al., 2003). In general the reliability of the resulting system can be highly correlated to the SNR of the input, depending on the environmental noise and the distance from speaker to microphones. In (Ramirez et al., 2005), more details are given on the problem, together with a good introductory survey of the audio event detection techniques explored more recently.

Also SLOC technologies have been deeply investigated and several different approaches are available in the literature (Wölfel and McDonough, 2009; Brandstein and Ward, 2001; Huang and Benesty, 2004). In general, SLOC algorithms are based on the estimation of the Time Differences Of Arrivals (TDOA) at two or more microphones, from which the source location is inferred by applying geometrical considerations. The Generalized Cross-Correlation Phase Transform (GCC-PHAT) (Knapp and Carter, 1976), is the most common technique for estimating the TDOA at two microphones. In multi-microphone configurations SLOC techniques based on acoustic maps, like the Global Coherence Field (GCF) (DeMori, 1998) also known as SRP-PHAT (Brandstein and Ward, 2001), are particularly effective in representing the spatial distribution of sources. Under the assumption that sources are sparse in time and space short-term spatio-temporal clustering has been successfully applied to the localization of multiple sources (Di Claudio et al., 2000; Lathoud and Odobez, 2007). Sequential bayesian methods and particle filtering (Arulampalam and Maskell, 2002; Vermaak and Blake, 2001; Lehman and Johansson, 2007) have also been experimented successfully on tracking of single as well as multiple sources (Fallon, 2008; Lee et al., 2010). Beside the above-mentioned methods, more recently approaches for Blind Source Separation (BSS), relying on Independent Component Analysis (ICA) (H. Sawada et al., 2003; Loesch et al., 2009) or on sparsity-aware processing of the cross-spectrum (Araki et al., 2009; Nesta and Omologo, 2011), have been applied to the estimation of the TDOA in presence of multiple sources (Brutti and Nesta, 2013).

1.1 Motivation

One of the main issues of the multi-room scenario typical of the domestic context, is that acoustic waves propagate from one room to another (e.g. through open doors), which represents an intrinsic cause of ambiguity on the location of each sound source, especially when concurring events can occur in different rooms. Furthermore, the environmental conditions of a domestic scene (e.g., background noise, interferes, noise sources, number of users, etc...) significantly vary over time, from very quiet conditions to very noisy and challenging situations, requiring algorithmic solutions capable of coping with such variability while preserving good performance. In DIRHA,

these challenges are tackled by distributing multiple microphones in the rooms of an apartment. This approach permits the implementation of effective SLOC solutions to identify the actual location of event generation as well as the development of robust strategies for event detection and speech recognition, for instance based on channel or model selection (Wolf and Nadeu, 2013; Sehr et al., 2010). The joint use of SLOC and SAD technologies is hence required in the addressed scenario in order to realize a multi-room SLOC and SAD. Although SAD and SLOC technologies have been widely investigated over the decades and several effective solutions are available in the literature, the peculiarities of the domestic scenarios pose significant challenges for these technologies. This fact motivated the creation of the DIRHA corpora and the definition of the SASLODOM evaluation tasks.

2 The DIRHA corpora

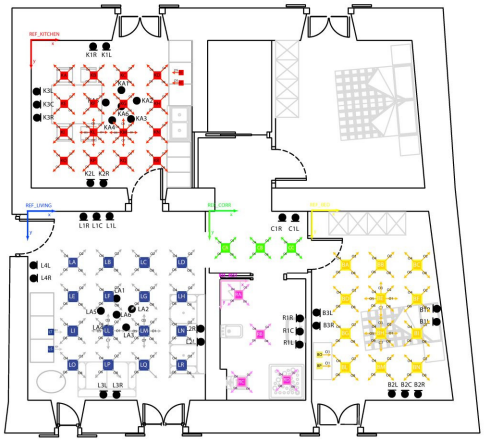


Figure 1: Layout of the apartment used for the collection of the DIRHA corpora. Circles indicate the microphone positions. Squares and arrows indicate the possible positions and orientations of acoustic events in the simulated corpus.

The general scenario addressed in the DIRHA project refers to a real automated apartment consisting of 5 rooms. In each room a set of microphones is deployed on the walls and the ceiling, as shown in Figure 1. 15 microphones are located in the Livingroom (bottom-left), 13 in the Kitchen (top-left), 7 in the Bedroom (bottom-right), 3 in the Bathroom (bottom-middle) and 2 in the Corridor (central). A star-shaped 6-microphone array is mounted on the ceiling of the Livingroom and of the Kitchen, where the majority of the speech events is expected to occur in every-day interactions. Overall 40 microphones monitor the house. For this target scenario, both simulated and real corpora of multi-channel multi-lingual acoustic data were created, in order to reproduce a variety of typical domestic scenes for experimental purposes (Cristoforetti et al., 2014). For each of the 40 microphones a 48 kHz/16 bit WAV audio file is available, fully synchronized and aligned at sample level with the other channels. Detailed annotations in terms of acoustic events, source positions and other information are also available. The corpora are publicly available upon request to the DIRHA consortium. The next sections provide a brief description of the two corpora. Table 1 summarizes the main differences between the simulated and real data collections.

	Real	Simulations
source	human	loudspeaker
movement	moving	static
system feedback	yes	no
background	quiet	various
noise source rate	low	high
overlapping events	no	yes

Table 1: Main differences between the real and simulated scenes.

2.1 The DIRHA SimCorpus

First of all, for a set of predefined positions and orientations (represented by squares and arrows in Figure 1) Room Impulse Responses (RIR) were measured for the 40 microphones by exciting the environment with long Exponential Sine Sweep (ESS) signals (Farina, 2000) reproduced by a loudspeaker. This procedure ensures high SNR and remarkable robustness against harmonic distortions (Ravanelli et al., 2012).

Speech events including sentences uttered by 120 speakers in 4 languages (Greek, German, Italian and Portuguese) were recorded using high-quality close-talking microphones and ensuring very high SNR and absence of artifacts. These sentences are typical commands for the domestic system, phonetically rich sentences and conversational speech. For what concerns "non-speech" events, they were selected from Logic Pro and from the Freesound[2] high-quality database, con-

[2]http://www.freesound.org/

sidering those sounds typical of domestic environments. Moreover, a selection of copyright-free radio shows, music and movies were used to simulate radio and television sounds. To increase the realism of the acoustic sequences, 21 common home-noise sources (shower, washing machine, oven, vacuum cleaner, etc.) were directly recorded by the distributed microphone network of the apartment.

Given the ingredients described above, the DIRHA SimCorpus (Cristoforetti et al., 2014) was created as a collection of acoustic scenes with a duration of 60 seconds. Each scene consists of real background noise, with random dynamics, to which a variety of localized acoustic and speech events are superimposed. Events occur randomly in time and in space, constrained on the grid of the predefined positions and orientations for which RIR measurements are available. The acoustic wave propagation from the sound source to each single microphone is simulated by convolving dry signals with the respective RIR.

Data set	Development	Test
Simul	40 scenes 40 min. 23.4% speech	40 scenes 40 min. 23.7% speech
Real	12 scenes 11 min. 9% speech	10 scenes 10 min. 30 sec. 17% speech

Table 2: Development and test material used in the SASLODOM task.

2.2 Real corpus

Besides the simulated scenes, a real data set was derived from excerpts of a Wizard-of-Oz data collection, resulting in 22 scenes, each one approximately 60 second long. Each real scene includes a human speaker uttering typical commands while moving within the Livingroom and the Kitchen. The background is rather quiet (in particular if compared to the simulated scenes), and the main noise of interference is the system output reproduced by the Wizard through a loudspeaker installed on the ceiling of the Livingroom or of the Kitchen (e.g., the replies of the system to the user commands). The reference signal of the system output is also made available.

2.3 Data used in the SASLODOM task

For the SASLODOM task a subset of the simulated data, consisting in 80 scenes in Italian, was considered. The scenes are selected in such a way that different degrees of complexity are covered. Notice that the language is probably not relevant for the addressed technologies.For what regards the real data, the full data set is used since it is relatively small and in Italian.

The data are evenly split in two sets for development and tests. Table 2 summarizes the amount of data used in the evaluation and the ratio between the total length of speech events over the full datasets duration.

3 The Task

Given the multi-room domestic scenario addressed in the DIRHA project, the goal of the SASLODOM task is, for each speech event, to:

- provide the corresponding time boundaries,
- determine the room where it was generated,
- derive the spatial coordinates of the speaker.

When considering a specific room, speech events occurring in other rooms must be discarded. Similarly, any other noise event must be neglected. In case a speech event occurring in a given room is associated by the system to another room, this will result in a false alarm and a deletion. Although speech and noise events may occur anywhere in the apartment, the evaluation considers only speech events generated in the Livingroom and Kitchen (i.e., speech events in other rooms must be discarded). This choice is motivated by the fact that a small number of microphones is available in the other rooms.

To allow the participation of laboratories without effective solution for SLOC, a subtask is defined where the localization stage does not require the estimation of the speaker coordinates but just the identification of the room where the event occurred (localization is implicit in the SAD component). This subtask is referred to as SAD.

4 System Evaluation

Reference speaker positions and speech activities are reported every 50 ms in a reference file, together with the annotation of other acoustic events occurring in the 5 rooms. The system under evaluation delivers, for each room and each scene, a

similar hypothesis file with a time resolution of at least 50 ms. If the time resolution of the hypotheses is higher, the evaluation tool averages the estimated coordinates.

In the evaluation step, the hypothesis sequence and the reference file are compared one each other. For each reference line, the closest (in time) hypothesis line is selected and one of the four events below is generated:

- Deletion: no hypothesis available for a given reference line (SAD);
- False Alarm: an hypothesis is produced when there is no speech activity in the targeted room (SAD);
- Fine error: the distance between the estimated source position and the reference is smaller than 50 cm;
- Gross error: the distance between the estimated source position and the reference is larger than 50 cm.

4.1 Metrics

Given the classifications listed above, a series of metrics is computed to characterize the performance of the system under evaluation:

- Time boundaries accuracy:
  - **Deletion Rate**: number of missing hypotheses over all speech frames.
  - **False Alarm Rate**: number of false alarms over all non-speech frames.
- Event-based Detection performance:
  - **Precision** of the SAD component.
  - **Recall** of the SAD component.
  - **F score**.

Systems are ranked according to the **Overall SAD Detection error**, defined as:

$$\text{SAD} = \frac{N_{\text{fa}} + \beta N_{\text{del}}}{N_{\text{nsp}} + \beta N_{\text{sp}}},$$

where N_{del}, N_{fa} are the total numbers of deletion and false alarms respectively, N_{sp} is the total number of speech frames, N_{nsp} is the total number of non-speech frames while $\beta = \frac{N_{\text{nsp}}}{N_{\text{sp}}}$ weights the contributions of false alarm and deletions. This weighting is necessary to avoid that results are biased due to the unbalanced distribution of speech and non-speech frames in the data (see Table 2). The SAD metric is equivalent to the Equal Error Rate in most of the cases. For a deeper understanding of the evaluation results, wherever possible the scores are reported in a disaggregated fashion, differentiating among cases in which there are noises in the targeted room, interferes (noise or speech) in another room, background noises.

The evaluation protocol includes also a set of metrics for the source localization tasks. Since none of the participants provided results on this problem they are not fully described here. They comprises: the average (bias) and RMS errors for fine and gross errors respectively as well as the ratio between the two categories (percentage of correct localization estimates).

It is worth mentioning that in an ASR perspective false alarms are less problematic than deletion as the rejection model offers an effective and practical way to deal with them. Therefore, it could make sense to give Deletions a higher weight in the overall SAD error rate computation. However, in the addressed context false alarms include also correct event associated to wrong rooms: this case would be detrimental for ASR and dialogue engines. This is the reason why the two rates are equally weighted.

4.2 Participants

As reported in Table 3, two laboratories participated in the evaluation, focusing on event detection and room selection only, and both participants submitted more than one system. The Spoken Language Systems Laboratory of the Instituto de Engenharia de Sistemas e Computadores Investigao e Desenvolvimento in Lisbon (INESC-ID L^2F) submitted three systems based on Multi-Layer Perceptron (MLP) and Major Voting Fusion (MVF) of the multiple channels. The three systems differ in the way the room selection is performed: MVF-MLP-NRS does not select the room while MVF-MLP-MRS and MVF-MLP-RRS adopt two slightly different procedures. The Multimedia Assistive Technology Laboratory - Dipartimento di Ingegneria dell'Informazione of the Universitá Politecnica delle Marche (MATeLab-DII) presented two approaches based on Deep Belief Networks (DBN) and Bidirection Long Short-Term Mem-

ory Recurrent Neural Networks (BLSTM) respectively. It must be mentioned that, although no SASLODOM specific data were used for system tuning, neither simulated nor real, the MLP models used by INESC-ID L^{2}F have been adapted on a rather large set of in-domain DIRHA data, not available to the other participant, which could give a significant improvement in the performance.

4.3 Results

Table 4 reports the evaluation results on the simulated corpus. Besides the official metrics the table reports the results also in terms of event-based metrics. The best performing system is "MVF-MLP-NRS" from INESC-ID L^{2}F which achieves a 7.7% error rate at frame level. However, this is obtained allowing events to occur in more than one room, which results in a considerable increase of false alarms and a significant reduction in the event-based metrics. In particular, the false alarm rate doubles in presence of events outside the target room. The reason why "MVF-MLP-NRS" performs better than the other two systems could be that the room selection scheme fails in several cases, in particular when noises outside the room occur. This fact confirms that the room selection problem is not a trivial task at all. In general all system submitted by INESC-ID L^{2}F handles properly the background noise, while a performance degradation is observed when events occurs outside the room. Note that the second best approach, which achieves a 9.5% overall error rate, has a very low precision despite acceptable false alarm and deletion rates: the reason could be in the generation of several short events. For both MATeLab-DII solutions background noise determines an increase of deletions (features are not observable) while noise events outside the rooms results in a higher false alarm rate (events are detected in the wrong room). It must be kept in mind that DNN solutions are penalized by the limited amount of training material.

4.4 Real Data

Table 5 reports the results on the real data. As expected the performance of the best systems is much higher than on the simulated data, thanks to the reduced amount of background noise and the absence of interfering sources. Furthermore, in the real data set events never overlap in time. In this case the best approaches are "MVF-MLP-MRS" and "MVF-MLP-RRS" of INESC-ID L^{2}F which outperform the solution without room selection. Given the easier conditions the room selection behaves properly and this provides a significant improvement to the performance. The methods proposed by MATeLab-DII performs considerably worse than on the simulated data, probably due to the limited amount of training material available.

5 Conclusions

The SASLODOM task at EVALITA 2014 addressed the problem of detecting and localizing speech event in a multi-room domestic scenario. The evaluation, based on real and simulated acoustic corpora collected within the EU DIRHA project, attracted two participants who focused on the SAD subtask. The submitted systems implement state of the art MLP and DNN solutions for the speech/non-speech classification task. The results confirm that the domestic scenario is extremely challenging and specific solutions based on multi-channel processing and room selection/localization are crucial to obtain satisfactory performance. In terms of absolute numbers, a very good accuracy is achieved on the real data.

Acknowledgements

This work has partially received funding from the European Union's 7th Framework Programme (FP7/2007-2013), grant agreement n. 288121-DIRHA.

Site ID	Full Name	Task	Runs
INESC-ID L^{2}F	Spoken Language Systems Laboratory Instituto de Engenharia de Sistemas e Computadores Investigao e Desenvolvimento Lisboa, Portugal	SAD	3
MATeLab-DII	Multimedia Assistive Technology Laboratory Dipartimento di Ingegneria dell'Informazione Universitá Politecnica delle Marche Ancona, Italy	SAD	2

Table 3: The participants of the SASLODOM task.

Lab	System	**SAD**	FA	Del	P	R	Fscore
INESC-ID L^{2}F	MVF-MLP-MRS	14.4	3.6	25.2	82.3	75.1	78.5
	MVF-MLP-RRS Sys2	11.8	5.4	18.2	73.4	79.2	76.2
	MVF-MLP-NRS Sys3	7.7	12.0	3.4	53.5	95.9	68.7
MATeLab-DII	BLSTM	12.1	11.9	12.3	30.6	98.6	46.5
	DBN	9.5	8.7	10.3	25.3	99.5	40.4

Table 4: Evaluation results on the simulated data.

Lab	System	**SAD**	FA	Del	P	R	Fscore
INESC-ID L^{2}F	MVF-MLP-MRS1	2.0	2.7	1.3	100	96.2	98.1
	MVF-MLP-RRS	2.0	2.7	1.3	100	96.2	98.1
	MVF-MLP-NRS	13.7	26.1	1.3	49.2	96.2	65.1
MATeLab-DII	BLSTM	19.7	33.7	5.6	22.5	98.7	36.7
	DBN	12.2	9.7	14.7	28.5	98.7	44.2

Table 5: Evaluation results on the real data.

References

Shoko Araki, Tomohiro Nakatani, Hiroshi Sawada, and Shoji Makino. 2009. Stereo source separation and source counting with MAP estimation with Dirichlet prior considering spatial aliasing problem. In *Proc. of the International Conference on Independent Component Analysis and Signal Separation*.

L. Armani, M. Matassoni, M. Omologo, and P. Svaizer. 2003. Use of a CSP-based voice activity detector for distant-talking ASR. In *EUROSPEECH*.

M. Arulampalam and S. Maskell. 2002. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), February.

R.L. Bouquin-Jeannes and G. Faucon. 1995. Study of a voice activity detector and its influence on a noise reduction system. *Speech Communication*, 16.

M. Brandstein and D. Ward. 2001. *Microphone Arrays*. Springer-Verlag.

A. Brutti and F. Nesta. 2013. Tracking of multidimensional tdoa for multiple sources with distributed microphone pairs. *Computer Speech And Language*, 27(3).

R Cai, L. Lu, A. Hanjalic, H. Zhang, and L. Cai. 2006. A flexible framework for key audio effects detection and auditory context inference. *IEEE Trans. on Audio, Speech and Language Processing*, 14(3).

W. Chu, W. Cheng, J. Wu, and J. Hsu. 2004. A study of semantic context detection by using SVM and GMM approache. In *Proc. of IEEE International Conference on Multimedia and Expo*.

L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmueller, and P. Maragos. 2014. The DIRHA simulated corpus. In *LREC*.

R. DeMori. 1998. *Spoken Dialogues with Computers*. Academic Press, London. Chapter 2.

E. Di Claudio, R. Parisi, and G. Orlandi. 2000. Multisource localization in reverberant environments by root-music and clustering. In *Proc. of IEEE conference on Acoustics, Speech, and Signal Processing*.

J. Dines, J. Vepa, and T. Hain. 2006. The segmentation of multichannel meeting recordings for automatic speech recognition. In *Proc. Int. Conf. on Speech Communication and Technology*.

M. Fallon. 2008. Multi target acoustic source tracking with an unknown and time varying number of targets. In *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, May.

A Farina. 2000. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *110th AES Convention*, February.

G. Guo and S. Li. 2003. Content-based audio classification and retrieval by support vector machines. *IEEE Trans. on Neural Networks*, 14(1).

H. H. Sawada, R. Mukai, and S. Makino. 2003. Direction of arrival estimation for multiple source signals using independent component analysis. In *Proceedings of ISSPA*.

Y. Huang and J. Benesty. 2004. *Audio Signal Processing for Next-Generation Multimedia Communication Systems*. Kluwer Academic Publishers.

J.C. Junqua, B. Mak, and B. Reaves. 1994. A robust algorithm for word boundary detection in the presence of noise. *IEEE Trans. on Speech and Audio Processing*, 2(3).

C. H. Knapp and G. C. Carter. 1976. The generalized correlation method for estimation of time delay. In *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, volume 24, pages 320–327.

G. Lathoud and J.M. Odobez. 2007. Short-term spatio-temporal clustering applied to multiple moving speakers. *IEEE Trans. on Audio, Speech and Language Processing*, 15(5), July.

Y. Lee, T.S. Wada, and Biing-Hwang Juang. 2010. Multiple acoustic source localization based on multiple hypotheses testing using particle approach. In *IEEE International Conference on Acoustics Speech and Signal Processing*.

E Lehman and A. Johansson. 2007. Particle filter with integrated voice activity detection for acoustic source tracking. *EURASIP Journal on Applied Signal Processing*.

B. Loesch, S. Uhlich, and B. Yang. 2009. Multidimensional localization of multiple sound sources using frequency domain ICA and an extended state coherence transform. *Proceedings of IEEE Workshop on Statistical Signal Processing*.

D. Macho, J. Padrell, A. Adad, J. McDonough, M. Wolfel, A. Brutti, M. Omologo, G. Potamianos, S. Chu, U. Klee, P. Svaizer, C. Nadeu, and J. Hernando. 2005. Automatic speech activity detection, source localization and speech recognition on the chil seminar corpus. In *Proc. of IEEE International Conference on Multimedia and Expo*.

F. Nesta and M. Omologo. 2011. Generalized State Coherence Transform for multidimensional TDOA estimation of multiple sources. *Audio, Speech, and Language Processing, IEEE Transactions on*.

T.V. Pham, M. Stadtschnitzer, Pernkopf F., and Kubin G. 2008. Voice activity detection algorithms using subband power distance feature for noisy environments. In *Proc. of Interspeech*.

J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro. 2008. Non-speech audio event detection. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*.

J. Ramirez, J.C. Segura, C. Benitez, A. De la Torre, and A. Rubio. 2005. An effective subband osf-based vad with noise reduction for robust speech recognition. *IEEE Trans. on Speech and Audio Processing*, 13(6), Nov.

M. Ravanelli, A. Sosi, M. Omologo, and Svaizer P. 2012. Impulse response estimation for robust speech recognition in a reverberant environment. In *EUSIPCO*.

A. Sehr, R. Maas, and W. Kellermann. 2010. Reverberation model-based decoding in the logmelspec domain for robust distant-talking speech recognition. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(7):1676–1691.

J. W. Shin, J.H. Chang, and N. S. Kim. 2010. Voice activity detection based on statistical models and machine learning approaches. *Computer Speech and Language*, page 515=530.

I. Trancoso, J. Portelo, M. Bugalho, J. da Silva Neto, and A. Serralheiro. 2009. Training audio events detectors with a sound effects corpus. In *Proc. of Interspeech.*

J Vermaak and A. Blake. 2001. Nonlinear ltering for speaker tracking in noisy and reverberant environments. In *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*.

M. Wolf and C. Nadeu. 2013. Channel selection measures for multi-microphone speech recognition. *Speech Communication*.

M. Wölfel and J. McDonough. 2009. *Distant speech recognition*. Wiley.

S.N. Wrigley, G.J. Brown, V. Wan, and S. Renals. 2005. Speech and crosstalk detection in multichannel audio. *IEEE Trans. on Speech and Audio Processing*, 13(1):84–91, Jan.

X.L. Zhang and J. Wu. 2013. Deep belief networks based voice activity detection. *IEEE Trans. on Audio, Speech, and Language Processing*, 21(4):679–710, April.

10.12871/clicit2014227

The L^{2}F system for the EVALITA-2014 speech activity detection challenge in domestic environments

Alberto Abad, Miguel Matos, Hugo Meinedo, Ramon F. Astudillo, Isabel Trancoso
INESC-ID/IST Lisbon, Portugal
{alberto.abad, jmatos, hugo.meinedo, ramon.astudillo, isabel.trancoso}@l2f.inesc-id.pt

Abstract

English. The INESC-ID's Spoken Language Systems Laboratory (L^{2}F) submission to EVALITA-2014 targets the problem of room-localized speech activity detection in multi-room domestic environments. The three proposed systems, which have been developed within the activities of the DIRHA project, combine multi-channel model-based speech classification with automatic room localization, based on spectral envelope distortion measures. The processing chain of the investigated approaches is composed of three basic stages: 1) multi-channel speech segmentation is carried out for each room, 2) speech segments detected at each room are time-aligned, and 3) a room assignment strategy is applied to each candidate speech event to determine in which room it was generated. The three submitted systems exploit the same speech/non-speech adapted model and the same channel combination strategy, while differing in the room localization strategy. Results obtained in the official EVALITA-2014 task confirm the effectiveness of the proposed methods. Particularly, in the case of real test data, F-scores of 98.1% are attained.

Italiano. *Il sistema sottomesso da INESC-ID Spoken Language Systems Laboratory (L^{2}F) affronta il probelma del rilevamento del parlato con relativa assegnazione ad una stanza in un tipico ambiente domestico caratterizzato da numerose stanze. I tre sistemi proposti, sviluppati nell'ambito del progetto DIRHA, combinano una prima classificazione del parlato, ottenuta attraverso un'elaborazione multi canale, con una selezione della stanza basata sulla distorsione dell'inviluppo spettrale. Il sistema e' costituito da tre componenti: 1) una segmentazione multi canale effettuata su ogni stanza; 2) i segmenti identificati sono allineati temporalemente; 3) una stanza viene assegnata ad ogni candidato. I tre sistemi adottano lo stesso modello di speech/non-speech e la stessa strategia nel combinare i canali, mentre si differenziano nel modo in cui viene selezionata la stanza da associare a ciascun evento. I risultati ottenuti sul task ufficiale di EVALITA-2014 confermano la convenienza dei metodi presentati. In particolare, sui dati reali i sistemi proposti raggiungo una F-score pari al 98.1%.*

1 Introduction

Speech activity detection of the acoustic input constitutes a crucial component in any voice-enabled application, providing important information to other system components, such as speaker localization, keyword spotting, automatic speech recognition, and speaker recognition, among others. In general, the quality of the segmentation information has a huge impact on the following speech processing components and its relevance is exacerbated for services that are required to work in an "always-listening" mode. This is the case of home automation applications. In fact, for such domestic scenarios, additional challenges affecting the performance of speech activity detection usually arise. First, microphones are normally located far from the source speaker in an environment that can be highly dynamic, noisy and reverberant. Second, in addition to detect "when" a speech activity has taken place, in multi-room environments it is important to decide "where" in the house such activity occurred.

The Speech Activity detection and Speaker

Figure 1: Block diagram of the speech/non-speech segmentation module.

LOcalization in DOMestic environments (SASLODOM) challenge, that is part of the EVALITA'2014 evaluation campaign, focuses on the detection and localization of speech events generated by users within the various rooms of a household. The scenario addressed in the task is the one of the DIRHA project (DIRHA, 2012), that is, an apartment monitored by 40 microphones, distributed on the walls and the ceiling of its five rooms. It encompasses typical situations observable in domestic contexts, in terms of speech input as well as of other acoustic events and background noise. For each speech event, the goal of the task is to: a) provide the corresponding time boundaries, b) determine the room where it was generated, and c) derive the spatial coordinates of the speaker. The task is evaluated in both simulated and real data sets in Italian, created by the DIRHA consortium. Additional details about the task, including guidelines, data, evaluation tools, details about the rooms and about the microphones are available in the SASLODOM task report (A. Brutti et al, 2014).

This report describes the L^{2}F speech activity detection (SAD) systems submitted to the SASLODOM challenge. The proposed systems have been developed within the activities of the DIRHA project.The complete room-localized SAD system is based on a three stage process. First, multi-channel speech segmentation is carried out for each room. Second, speech segments detected at each room are time-aligned in order to identify speech events that are likely to be the same. Third, a room assignment strategy is applied to each candidate speech event to determine in which room it was generated.

2 The L^{2}F multi-room SAD systems for domestic environments

The L^{2}F multi-room SAD systems have been developed in the context of the DIRHA project. This section provides details on different approaches investigated and evaluated using DIRHA data.

2.1 The DIRHA SimCorpus

The DIRHA SimCorpus (L. Cristoforetti et al, 2014) is a multi-microphone and multi-language database containing simulated acoustic sequences derived from the microphone-equipped apartment located in Trento (Italy) (M. Ravanelli et al, 2014). In this work, the development set of the DIRHA SimCorpus has been used to adapt the speech/non-speech model that is part of the SAD module (more details in section 2.2). On the other hand, the test set of the European Portuguese DIRHA SimCorpus is used to assess the different methods under study.

2.2 Baseline MLP-based SAD detector

The core module of the L^{2}F systems is a model-based speech/non-speech classifier. This module is composed by several blocks, as depicted in Figure 1. The first one, designated as feature extraction, performs acoustic parametrization of the audio signal, extracting 12th order perceptual linear prediction (PLP) coefficients plus signal frame energy, all appended by their first temporal derivatives, thus yielding 26-dimensional acoustic features. These are subsequently passed to the classification block, which is implemented using an artificial neural network of the multi-layer perceptron (MLP) type (Meinedo, 2008). The baseline neural classifier was trained using 50 hours of TV Broadcast News and 41 hours of varied music and sound effects (in order to improve the representation of non- speech audio signals). The output of the trained neural classifier represents the probability of the audio signal containing speech. The following block smooths this probability using a median filter over a small window. The smoothed signal is then thresholded and analysed using a time window (t_{min}). The final block is a finite state machine that consists of four possible states ("probable non-speech" "non-speech", "probable speech", and "speech"). More details can be found in (A. Abad et al, 2013).

3 Baseline for distant speech recognition in Portuguese

3.1 Improvements to the baseline SAD

The aim of this section is to improve the baseline SAD module. For that purpose, we define a new task that consists of detecting speech events occurring in a specific room and ignoring the speech events that occur in the other rooms. We refer to this task as the "isolated-room" SAD task. Notice that this is not the targeted task in the SASLODOM challenge. Nevertheless, this "isolated-room" SAD task permits the assessment of the proposed systems ignoring the errors due to cross-room speech insertions, which is a particularity of multi-room environments. In this section, the DIRHA SimCorpus for European Portuguese (PT) was used for testing.

3.1.1 MLP adaptation

The MLP model described previously is not at all adjusted to the acoustic environments targeted at DIRHA. A reasonable solution for this problem is to retrain or adapt the MLP based classifier using appropriate data, that is, data more similar to the test conditions. To evaluate the feasibility of this approach, the baseline MLP classifier was adapted using three development sets from the DIRHA SimCorpus, namely the ones in Italian (IT), European Portuguese (PT), and Greek (GR). As described in (M. Ravanelli et al, 2014), the simulated data correspond to microphones located in five rooms of the apartment. For each room, a specific microphone was chosen. A total of 1125 audio files from the 3 languages, 5 rooms, and 75 recorded simulations were used in the adaptation, of which 750 for training and the remaining 375 to validate the model. The MLP was fully adapted using a single epoch of back-propagation, with a much smaller learning step than the one used for the initial model training.

3.1.2 Multi-channel combination

In addition to the adaptation of the speech/non-speech model, improved segmentation for each room is obtained by exploiting all the microphones available in the apartment. We explore two methods of multi-channel combination: Majority Voting Decision Fusion (MVF) and Posterior Probability Fusion (PF).

Majority Voting Decision Fusion (MVF) In the MVF method, the baseline speech/non-speech segmentation module is first run individually for each channel of the house. Then, the resulting segmentations from all the channels of a specific room are aligned to detect candidate speech events. Due to the possible different propagation delays from the speech source to the several microphones, a tolerance of 1 second is given to this alignment process. Then, if more than half of the microphones of a specific room detect a speech event candidate, the system considers that there was speech in that room in that time interval.

Posterior Probability Fusion (PF) In the PF method, the posterior probabilities obtained by the MLP classifier for each channel of a specific room are combined before applying the median filter. The combination rule is simply the mean of the probabilities provided by the MLP. Then, the same finite state machine adopted in the single-channel case is used to obtain the room segmentation based on these averaged probabilities.

3.1.3 "Isolated-room" SAD task results

The results of the distinct approaches are presented in Table 1. In the mono-channel system, a representative microphone was chosen for each room. Observing the speech recall values of Table 1, it can be seen that the MLP unadapted system (*MLP-Baseline*) rejects a very high percentage of speech. After adaptation of the network classifier with in-domain data (*MLP-DIRHA*), speech recall increases to around 80%, while maintaining a high non-speech detection precision. Regarding multi-channel combination approaches, generalized improvements (F-score) are attained with respect to the mono-channel approach. There are no significant differences between the two multi-channel methods.

3.2 Room-Localized SAD

In this section, we focus on the SASLODOM task, that we refer to as "room-localized" SAD task. Notice that in contrast to the previous section, the detected speech segments which originated in other rooms are considered as insertion errors and affect the performance of the evaluated systems. Table 2 presents the results achieved by the SAD systems previously described when evaluated in the "room-localized" task. As it can be observed, performances greatly decrease compared to the ones reported in Table 1. This is due to the high rate of detected speech segments actually occur-

System [channel + MLP model]	speech			non-speech			total
	Prec.	Recall	F-score	Prec.	Recall	F-score	Acc.
1c + MLP-Baseline	99.7	54.7	70.6	95.2	100	97,5	95.4
1c + MLP-DIRHA	70.8	81.0	75.5	97.8	96.3	97.0	94.7
MVF + MLP-DIRHA	74.2	80.7	77.3	97.8	96.9	97.3	95.2
PF + MLP-DIRHA	76.1	79.9	77.9	97.7	97.2	97.5	95.5

Table 1: Performance (%) of the "isolated-room" speech activity detection task with the European Portuguese DIRHA SimCorpus test set using different MLP classifiers with single-channel and multi-channel combination approaches.

System [channel + MLP model]	speech			non-speech			total
	Prec.	Recall	F-score	Prec.	Recall	F-score	Acc.
1c + MLP-DIRHA	26.1	81.6	39,5	98.2	81.1	88.8	81.1
MVF + MLP-DIRHA	26.5	81.4	40.0	98.2	81.5	89.1	82.5
PF + MLP-DIRHA	27.5	80.4	41.0	98.1	82.7	89.7	81.5

Table 2: Performance (%) of the "room-localized" speech activity detection task with the European Portuguese DIRHA SimCorpus test set using different MLP classifiers with single-channel and multi-channel combination approaches.

ring in a different room. These results show the inadequacy of the proposed approaches for the targeted task.

3.2.1 Strategies for room detection

In order to address the cross-room detection problem, we propose to combine conventional SAD approaches with automatic room detection methods. The proposed method consists of a three-step process as follows:

1. Obtain automatic segmentation for each room using any of the previously described methods. With this operation, we obtain a set of speech candidate segments for each room.

2. Align speech candidate segments of all rooms with a tolerance of 1 second. This is done to match events that are likely to be the same ones, but that are simultaneously detected at different rooms.

3. Decide to which room every speech candidate segment belongs using the information provided by an automatic room detector.

From the various room-detection methods studied, the ones based on envelope variance (EV) distortion measures (M. Wolf and C. Nadeu, 2010) were chosen, because they present the best trade-off between computational load and performance for an environment with noise and reverberation. In this work, the detected room corresponds to the room of the microphone with the highest EV measure in the time interval of the candidate speech segments. In practice, we have explored two methods of integrating the segmentation information and the room localization information:

- *Restricted room selection (Restricted-RS)* The rooms in which the speech event may happen are restricted to those rooms that actually detected that hypothesised segment.

- *Matched room selection (Matched-RS)* Automatic room detection is not restricted and any room may be selected for each hypothesised speech segment. However, if the automatically selected room does not match any of the rooms that actually detected the hypothesized segment, then that candidate segment is disregarded.

In practice, the difference between the two methods is that in the first case, all aligned candidate segments are assigned to one room (and removed from any other room in which the same candidate is detected), while in the second case, there may be candidate segments that are disregarded and not assigned to any room. Consequently, for the second approach, one may expect an increase of the precision in exchange for a drop in the recall performance.

Room selec. approaches	System [channel + MLP model]	speech			non-speech			total
		Prec.	Recall	F-score	Prec.	Recall	F-score	Acc.
Restricted-RS	1c + MLP-DIRHA	43.2	65.9	52.2	97.1	92.9	95.0	90.9
	MVF + MLP-DIRHA	46.4	65.3	54.3	97.1	93.8	95.4	91.7
	PF + MLP-DIRHA	46.9	65.6	54.7	97.1	93.9	95.5	91.8
Matched-RS	1c + MLP-DIRHA	73.2	59.5	65.7	96.7	98.2	97.5	95.3
	MVF + MLP-DIRHA	75.2	59.6	66.5	96.7	98.4	97.6	95.5
	PF + MLP-DIRHA	74.9	59.8	66.5	96.8	98.4	97.6	95.4

Table 3: Performance (%) of the "room-localized" speech activity detection task with the European Portuguese DIRHA SimCorpus test set using applying single-channel and multi-channel fusion approaches combined with two different room-localization approaches based in EV.

Test data	System [channel + MLP model + RS]	O-SAD	FA	DR	Prec.	Recall	F-score
Simulated	MVF + MLP-DIRHA + Non-RS	7.7	12.0	3.4	53.5	95.9	68.7
	MVF + MLP-DIRHA + Restricted-RS	11.8	5.4	18.3	73.4	79.2	76.2
	MVF + MLP-DIRHA + Matched-RS	14.4	3.6	25.2	82.3	75.1	78.5
Real	MVF + MLP-DIRHA + Non-RS	13.7	26.1	1.3	49.2	96.2	65.1
	MVF + MLP-DIRHA + Restricted-RS	2.0	2.7	1.3	100	96.2	98.1
	MVF + MLP-DIRHA + Matched-RS	2.0	2.7	1.3	100	96.2	98.1

Table 4: Performance results (%) of the L^2F speech activity detection systems submitted to the SASLODOM challenge in the simulated and real data test sets in terms of the official task evaluation metrics: Overall SAD performance (O-SAD), false alarm rate (FA), deletion rate (DR), Precision (Prec), Recall and F-score.

3.2.2 "Room-Localized" SAD task results

Table 3 presents the results obtained for the two integrated approaches that combine speech activity detection and room localization. Comparing these results with the ones obtained with the systems that do not incorporate any room assignment strategy (Table 2), we can observe a great improvement in the precision performance of speech. On the other hand, there is also a considerable drop in the recall performance. However, we can see that the incorporation of room localization increases the system performance about 25% for the best method in terms of F-score. These results seem to demonstrate the convenience of the methods proposed that combine segmentation with room localization.

Regarding the room-assignment strategies, the recall is higher for the *Restricted-RS* approach, as expected, because all candidate segments are always assigned to one room. On the other hand, also as expected, the precision is very low when compared to the *Matched-RS* approach. In general, the second approach achieves a better generalised performance (F-score).

4 The L^2F SASLODOM 2014 submission

Three different systems have been submitted to the EVALITA-SASLODOM 2014 challenge. The three systems differ in the room selection strategy integrated: no room selection (*Non-RS*), restricted room selection (*Restricted-RS*) and matched room selection (*Matched-RS*). The three systems share the same MLP classifier adapted with in-domain data (MLP-DIRHA), since it showed remarkable improvements with respect to the baseline classifier in the experiments with the DIRHA SimCorpus. Moreover, given that no significant performance differences were observed regarding multi-channel combination methods, majority voting fusion (MVF) approach was applied in all cases. It is worth noting that system tuning has not been conducted to adapt to the particular characteristics of the SASLODOM data.

Table 4 shows the official performance results obtained by the submitted systems in the simulated and real data test sets. According to these results, the trends of the different systems are as expected: the highest recall/lowest precision is achieved by the system that does not incorporate

room detection strategies, while the *Matched-RS* is the room assignment strategy that provides highest precision in exchange for a moderate recall drop. Regarding F-score metrics, the *Matched-RS* approach is the best performing one. Comparing the *Simulated* results to the ones reported in the previous section, two relevant differences can be noticed. First, the general performance is considerably better: F-scores increase from 40.0%, 54.3% and 66.5% to 68.7%, 76.2% and 78.5%, for each of the three submitted systems respectively. Second, the performance differences between the three systems are considerably reduced. A possible explanation for these two observations may be the reduced amount of cross-room detected speech events in the SASLODOM data when compared to the DIRHA data. However, this is only an hypothesis that needs to be further investigated and there may be other explanations for the observed phenomena. Finally, it is worth highlighting the extremely good performances with real data (F-score 98.1%) achieved by the proposed approaches incorporating automatic room detection information. Note that these methods allowed for a drastic precision increase, from 49.2% to 100%, while keeping the recall constant at 96.2%. These figures show that each candidate speech segment is in fact simultaneously detected at the two rooms. However, the room assignment strategy based on EV is able to perfectly determine the correct room where each speech event is generated. This result confirms the effectiveness of the EV distortion metric for channel and room selection with real data.

Acknowledgements

This work was partially supported by the European Union, under grant agreement FP7-ICT-2011-7-288121, and by the Portuguese Foundation for Science and Technology, through project PEst-OE/EEI/LA0021/2013 and grant number SFRH/BPD/68428/2010. The authors would like to thank to their colleagues in the DIRHA consortium and to the organizers of the EVALITA-SASLODOM 2014 challenge.

References

DIRHA project. 2012. `http://dirha.fbk.eu/`.

A. Brutti et al. 2014. "SASLODOM: Speech Actvity detection and Speaker LOcalization in DOMestic environments," in *Proceedings of Evalita 2014.*. Pisa University Press, 2014.

L. Cristoforetti et al. 2014. "The DIRHA simulated corpus," in *Proc. LREC 2014*

M. Ravanelli et al. 2014. "DIRHA-simcorpora I and II," *Deliverables 2.1, 2.3, 2.4, DIRHA Consortium.*

H. Meinedo. 2008. "Audio pre-processing and speech recognition for BroadcastNews," Ph.D. dissertation, IST, Lisbon, Portugal.

A. Abad et al. 2013. "Multi-microphone front-end," *Deliverable D3.2, DIRHA Consortium*.

M. Wolf and C. Nadeu. 2010. "On the potential of channel selection for recognition of reverberated speech with multiple microphones," in *Proc. Interspeech 2010*:80–83.

10.12871/clicit2014228

Neural Networks Based Methods for Voice Activity Detection in a Multi-room Domestic Environment

Giacomo Ferroni, Roberto Bonfigli, Emanuele Principi, Stefano Squartini, and Francesco Piazza
Department of Information Engineering, Università Politecnica delle Marche
Via Brecce Bianche, 60131, Ancona, Italy
{g.ferroni,r.bonfigli,e.principi,s.squartini,f.piazza}@univpm.it

Abstract

English. Several Voice or Speaker Activity Detection (VAD) systems exist in literature. They are indeed a fundamental part of complex systems that deals with speech processing. In this work the authors exploit neural network based VAD to address the speaker activity detection in a multi-room domestic scenario. The goal is to detect the voice activity in each of the two target rooms in presence of other sounds and speeches occurring in other rooms and outside. A large dataset recorded in a smart-home is provided and interesting results are obtained.

Italiano. *Un rilevatore di attività vocale (Voice Activity Detector, VAD) costituisce una delle parti fondamentali di sistemi più complessi che operano con segnali vocali. Il presente lavoro applica VAD basati su reti neurali per il rilevamento del parlato in uno scenario domestico multimicrofono. Lo scopo è quello di rilevare l'attività vocale presente nelle due stanze di riferimento in presenza di altri suoni e parlatori in altre stanze o all'esterno. Le prestazioni sono state valutate su un ampio dataset ed i risultati ottenuti sono interessanti.*

1 Introduction

Voice Activity Detection (VAD) is a non-trivial task representing one of the fundamental steps of many complex systems like Automatic Speech Recognition (ASR) (Rabiner and Juang, 1993). This work concerns the development and the evaluation of advanced VADs applied in domestic environments[1] (Principi et al., 2013). A large dataset is provided by the DIRHA EU project and it is composed of several scenes recorded using 40 microphones installed in five rooms of a smart-home (Cristoforetti et al., 2014). The approaches presented hereby are based on machine learning techniques, in particular, the first approach exploits the Deep Belief Network (DBN), a neural network obtained by stacking several Restricted Bolzmann Machines (RBMs) whilst the second approach is based on a bidirectional Long Short-Term Memory (LSTM) recurrent neural network. The proposed VADs at their current development stage have been submitted and their performance have been assessed at the Speech Activity detection and Speaker LOcalization in DOMestic environments (SASLODOM) task, part of EVALITA 2014[2].

The reminder of this technical report is structured as follows. A brief overview of the task dataset and an overall description of the proposed systems is given in the next two Sections. Section 4 describes the experimental setup while Section 5 shows the obtained results and Section 6 concludes the article.

2 SASLODOM 2014 dataset

The dataset provided by the DIRHA project refers to an apartment monitored by 40 microphones installed on the walls and the ceiling of its five rooms (cf. Figure 1). The target rooms in which the speech activity has to be detected is the kitchen (top-left) and the livingroom (bottom-left). The dataset is composed of two kind of sets named *Simulated* and *Real*. The first one is composed of 80 scenes 60 seconds long and they consist of a set of utterances and other acoustic events, including a variety of background noises, produced in different rooms and positions. The Real dataset is composed of 22 total scenes having different durations. They are composed of moving speaker utterances and system audio messages played through a ceiling loudspeaker. In these scenes the background

[1] The proposed systems are currently under development.

[2] http://www.evalita.it/2014

noise is low and the speakers are located only in the kitchen and livingroom.

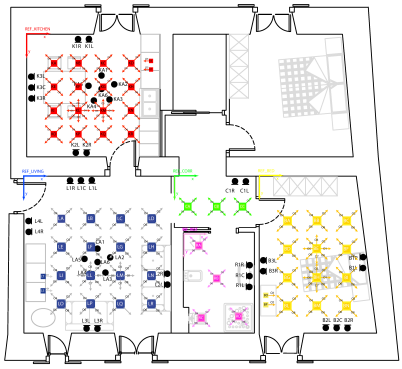


Figure 1: Layout of the experimental set-up for simulated data.

3 Overall description

The overall block scheme of the proposed approaches is depicted in Figure 2. The acquired input audio signals, coming from one or more microphones, is fed to the *feature extraction* block which aims to transform the raw audio data into a well-defined feature space (cf. Section 3.1). The feature matrix is then used as input for the *speech/non-speech* classifier. Finally a post-processing stage leads to the final decision.

3.1 Feature Extraction

Different types of features are extracted from raw audio data after down-sampling it to 16 kHz. The feature sets are normalised following the min-max method:

$$\bar{x}_l = \frac{x_l - x_{\min}}{x_{\max} - x_{\min}}, \quad (1)$$

where

$$x_{\min} = \min_{1 \le l \le L}(x_l), \quad x_{\max} = \max_{1 \le l \le L}(x_l), \quad (2)$$

x_l is an element of the feature vector at the frame index l and L is the total number of frame in the dataset. The complete list is shown in Table 1 whilst, the next sections provide a detailed description.

3.1.1 Mel-Frequency Cepstral Coefficient

The MFCC (Davis and Mermelstein, 1980) is a well-known set of features widely employed in audio applications (e.g., speech, music, etc.). Accordingly with HTK target kind (Young et al., 1997), two set of MFCC-based feature have been extracted: MFCC12_0_D_A and MFCC12_0_D_Z.

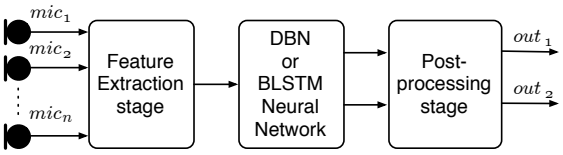


Figure 2: General block scheme of the proposed VADs.

Name	# features
MFCC12_0_D_Z *	26
MFCC12_0_D_A *	39
EVM_wH	1
PITCH *	1
WCLPE	24
RASTAPLP_0_D_A *	54

Table 1: List of features and their dimensionality. The * indicates that the features are extracted using openSMILE toolkit (Eyben et al., 2013).

The former is composed of 13 cepstral coefficients, 0-12, plus their first and second derivatives, Δ and $\Delta\Delta$ whilst the latter differs in the features mean normalisation and in the absence of the second order derivative. Both are extracted using a frame size of 25 ms at a frame rate of 100 fps.

3.1.2 Envelope-Variance measure

This feature relies on the signal intensity envelope smoothing introduced by the reverberation, thus, the dynamic range of a reverberated signal may be reduced (Houtgast and Steeneken, 1985). The extraction process have been slightly modified in order to achieve a temporal evolution. The original version (Wolf and Nadeu, 2014) defines a set of sub-band envelopes as the time sequences of non-linearly compressed filter-bank energies (FBE). Similarly to MFCC computation, the speech signal frame energies is computed and the mean value is subtracted in the log domain from each sub-band:

$$\hat{x}(k,l) = \exp[\log(x(k,l)) - \mu_x(k)], \quad (3)$$

where $x(k,l)$ is the sub-band time sequence, k is the band index, l is the frame index and $\mu_x(k)$ is the k-th band mean value estimated along the entire speech sub-band signal. The variance of a compressed version of Eq. (3) is obtained as follow:

$$V(k) = var[\hat{x}(k,l)^{1/3}]. \quad (4)$$

To obtain a time-varying version of Eq. (4), we compute the variance using a window W shifted

along each sub-band time sequence:

$$EVM(k,l) = var[\hat{x}(k,m)^{1/3}], \qquad (5)$$

where the variance is calculated considering a portion of $\hat{x}(k,m)$ identified by $-\frac{W}{2} + l \leq m \leq \frac{W}{2} + l$. Finally, a hard weighting function is applied to emphasise the voiceband frequencies and to discard the others contents. We use $p = 40$ mel sub-bands and a windows size of 400 ms leading to the EVM_wH set.

3.1.3 Pitch

The pitch feature is extracted accordingly to the Sub-Harmonic-Summation (SHS) method (Hermes, 1988). It computes N_f shifts of the input spectrum along the log-frequency axis, each of them is scaled due to a compression factor and summed up leading to a sub-harmonic summation spectrum. Standard peak picking and a quadratic curve fitting interpolation are applied to identify the F_0 value. They are extracted using a frame size of 50 ms sampled every 10 ms.

3.1.4 RASTA-PLP

This feature set is the standard RASTA-PLP set (Hermansky, 1990) composed of 18 cepstral coefficients including the 0-th one plus their first and second derivatives. They are extracted using a frame size of 25 ms sampled every 10 ms.

3.1.5 WC-LPE Feature

The Wavelet Coefficient (WC) and Linear Prediction Errore (LPE) feature set is based on a sub-band multi-resolution representation due to the exploitation of the Discrete Wavelet Transformation of the input. A set of Linear Prediction Error Filters (LPEFs) is then applied to each sub-band in order to extract the Forward Prediction Errors (FPE). The latter, the WCs and their first average derivatives constitute the feature set presented in (Marchi et al., 2014). To guarantee a frame alignment with respect to other feature sets, the reference frequency has been set to 100 Hz.

3.2 Deep Belief Network

The DBN is well-defined in (Deng, 2012) as a probabilistic generative models composed of multiple layers of stochastic, hidden variables. The top two layers have undirected, symmetric connections between them. The lower layers receive top-down, directed connections from the layer above. A DBN is built by a stack of Restricted

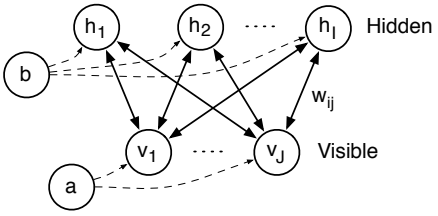


Figure 3: Restricted Boltzmann Machine.

Boltzmann Machines (RBMs) and the interest in this generative model began to increase since the introduction of an efficient layer-by-layer unsupervised training algorithm, also called pre-training (Hinton et al., 2006). DBNs are typically used to initialise the weights of a Multi-Layer Perceptron (MLP) neural network, especially when the MLP is composed of many layers (i.e., deep neural network, DNN). Following this initialisation, a standard back-propagation fine-tunes the network leading to much better results than that achieved by randomly initialise the MLP. When DBN in exploited for initialisation of a DNN, the obtained network is called DBN-DNN.

RBMs are composed of one layer of Bernoulli stochastic hidden units $\mathbf{h}$ and one layer of Bernoulli or Gaussian stochastic visible units $\mathbf{v}$, where $\mathbf{h}$ and $\mathbf{v}$ are the vector of hidden and visible unit values. With respect to Boltzmann Machines, RBMs have not hidden-to-hidden and visible-to-visible connections. Figure 3 shows a RBM with I visible units and J hidden units, w_{ij} indicates the weights between i-th visible unit v_i and j-th hidden unit h_j, and b_i and a_j are respectively the bias terms for visible and hidden layers. Following (Hinton, 2010), a RBM can be easily trained by means of Contrastive Divergence (CD-1) algorithm which allows to compute the approximation of the gradient of the log likelihood $\log p(\mathbf{v};\theta)$, where θ is the model parameters, by exploiting a full step of the Gibbs sampling method. A full step consists in sampling $\mathbf{h}_0$ from $\mathbf{v}_0$, then sampling $\mathbf{v}_1$ from $\mathbf{h}_0$ and, finally sampling $\mathbf{h}_1$ from $\mathbf{v}_1$. Hence, the weights update rule for the RBM is:

$$\Delta w_{ij} = \epsilon[\langle v_1 h_1 \rangle - \langle v_0 h_0 \rangle], \qquad (6)$$

where ϵ is the learning rate and the vector of visible units $\mathbf{v}_0$ are initialised using the input data.

In the stacking procedure, the RBMs are trained using the CD-1 algorithm layer by layer leading to a DBN as shown in Figure 4. Firstly RBM_1 is pre-

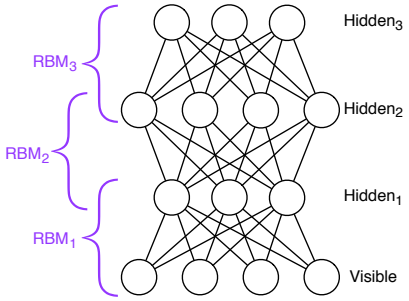


Figure 4: Deep Belief Network obtained by stacking three RBMs.

trained, then the hidden unit activation probabilities of RBM_1 became the visible units of RBM_2 and the pre-training algorithm is applied to RBM_2. Finally the hidden unit activation probabilities of RBM_2 became the visible units of RBM_3 which is pre-trained. This process proceeds iteratively for each layer in the network. It is important to note that this training procedure is unsupervised, thus, it does not require the targets or labels knowledge. For classification tasks, the pre-training is followed by a supervised training algorithm (e.g., back-propagation) which, on the contrary, exploits the targets to fine-tune the network weights.

3.3 Bidirectional LSTM-RNN

A BLSTM-RNN is a recurrent neural network in which the usual non-linear neurons (i.e., sigmoid function) are replaced by the long short-term memory blocks.

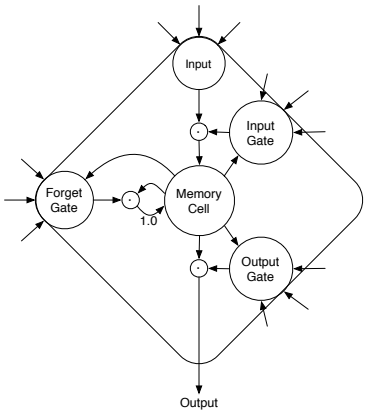


Figure 5: Long Short-Term Memory block.

The LSTM block is composed of one or more self connected linear memory cells and three multiplicative gates, as shown in Figure 5. The memory cell maintains the internal state for a long time through a constant weighted connection (i.e., 1.0). The content of the memory cell is controlled by the multiplicative input, output and forget gates which act respectively as the memory write, read and reset operations. More details can be found in (Hochreiter and Schmidhuber, 1997; Graves, 2012).

The recurrent nature of the network allows a kind of *memory* in the network internal state which is exploited to compute the output of the network. To deal with the future context, an elegant solution is to duplicate the hidden layers and connect them to the same input and output. The input values and corresponding output targets are thus given in a forward and backward direction. This network architecture is called Bidirectional LSTM-RNN (BLSTM-RNN).

4 Experimental Setup

The given dataset has been divided as provided by the SASLODOM 2014 organisers:

- **Development Set**: 40 scenes from the Simulated set and 12 scenes from the Real set.
- **Test Set**: 40 scenes form the Simulated set and 10 scenes from the Real set.

The Test Set has been provided to the participants at the end of the development phase in order to evaluate the performance, hence the feature selection, the network parameters identification and the post-processing variables tuning have been computed by means of a 10-fold cross validation over the Development Set.

4.1 DBN-VAD

The proposed DBN-VAD (cf. Figure 2) has two different configurations. In particular, the feature set and the network topology are different due to the diverse nature of the Simulated and Real sets. The feature set employed with the simulated dataset is composed of 106 coefficients/frame for each microphone: MFCC12_0_D_Z, EVM_wH, PITCH, WC-LPE and RASTAPLP_0_D_A. The network has 212 input units, two hidden layers of, respectively, 20 and 10 units and an output layer of two units, one for each target rooms. We refer to this configuration as DBN-VAD_S. On the other hand, both the feature set and the network size for the real dataset are smaller: 27 coefficients/frame MFCC12_0_D_Z and PITCH, and 57 inputs units, two hidden layers of 10 and 5 units and two output units. We refer to this configuration as DBN-VAD_R.

Both the configurations exploits two microphones installed on the kitchen wall (i.e., K2L) and on the livingroom wall (i.e., L1C). The choice of these two microphones relies on their position (cf. Figure 1) and also as a result of intensive tests conducted on several microphone pairs.

The DBN-VAD$_{S|R}$ pre-training consists in 1000 iterations using a mini-batch size of 100 frames and a step-ratio of 0.1. The learning rate is obtained dividing the step-ratio by the size of the training set leading to a value close to 4×10^{-7}. The fine-tuning training has the same parameters.

4.2 BLSTM-VAD

The second proposed VAD is BLSTM-based (cf. Figure 2) and exploits the two microphones used with the DBN-VAD (i.e., K2L and L1C). This VAD employs a different feature set composed of MFCC12_0_D_A, PITCH and WC-LPE leading to a total feature space of 64 coefficients per frame per microphone. The final network topology is composed of four hidden layers (i.e., two for each direction due to bi-directionality) with 40 and 20 LSTM units for each direction. The input layer has 128 units while the output layer has only one unit. Indeed, for this VAD approach, better performance has been achieved using one network for each room.

For BLSTM-VAD training, the CURRENNT toolkit (Weninger et al., 2014) is used. In particular, supervised learning with early stopping is used. Standard gradient descend with back propagation of the output errors is used to iteratively update the network weights. The latter are initialized by a random Gaussian distribution with mean 0 and standard deviation 0.1.

4.3 Post-processing

A post-processing of the network output is needed in order to handle slow transition from speech to non-speech. This technique is commonly named *hangover* and a number of different implementation have been developed. The simplest implementation, used in this work, exploits a counter. In particular, a threshold value is fixed and if at least two consecutive network outputs are above the threshold, the counter is reset to a predefined value (equal to 8). On the contrary, when the network output is below the threshold, the counter is decreased by 1 and the actual frame is classified as non-speech only if the counter value is zero.

5 Results

The result published by SASLODOM 2014 organisers are shown in this section.

5.1 Performance metrics

The metrics used to assess the VAD performance are:

- Deletion Error Rate (DER): number of missing detection over all speech frames.
- False Alarm Rate (FAR): number of false detection over all non-speech frames.
- Overall Speaker Activity Detection error (SAD): global metric defined as:

$$\text{SAD} = \frac{N_{fa} + \beta N_{del}}{N_{nsp} + \beta N_{sp}}, \tag{7}$$

  where N_{del}, N_{fa} are the total number of deletions and false alarms respectively, N_{sp} and N_{nsp} are the total number of speech and non-speech frames. The term $\beta = \frac{N_{nsp}}{N_{sp}}$ acts as regulator term for the unbalance of the class non-speech with respect to the speech one.

Table 2 shows the performance achieved by the proposed VADs with respect to the Test Set. The proposed VADs at their current development stage are characterised by moderate performance with respect to the Real dataset. This fact is due to the *raw* approach that authors decided to undertake as first step. In particular, the data-driven nature of our VADs does not exploit higher level information to finalise the decision. For instance it could be possible to exploit the envelope-variance measure (cf. Eq. (4)) to perform a channel selection and hence further post-processing the network decisions. This solution would reasonably improve the performance on Real dataset. Indeed, the absence of noise in its scenes leads to a high accuracy of the channel selection measure. Performance against the Simulated data are significantly better due to the grater dimension with respect to the Real data.

6 Conclusion

The proposed VADs exploit DBN-DNN and BLSTM-RNN neural networks in order to detect the speaker activity in a multi-room scenario. Indeed, the task goal is the detection of when and where a human is talking with respect to target rooms. Hence, the system is required to be robust

VAD	Simulated data			Real data		
	DER (%)	FAR (%)	SAD (%)	DER (%)	FAR (%)	SAD (%)
DBN-VAD$_{S\|R}$	10.3	8.7	9.5	14.7	9.7	12.2
BLSTM-VAD	12.3	11.9	12.1	5.6	33.7	19.7

Table 2: Result assessed against the Test Set.

and reliable in a noise environment and a multiple speaker scenario. Furthermore, the VAD is also required to identify in which room, kitchen or livingroom, the speaker is actually talking discarding other speaker(s) in other room(s). The performance of the proposed approaches have been assessed on the SASLODOM-EVALITA 2014 task. Further intensive test sessions focused to preprocess the multiple microphone signals available and to the evaluation of deeper networks represent future efforts. Moreover, due to the so-called *curse of dimensionality*, better performance are expected by the exploitation of the whole DIRHA dataset.

Acknowledgment

The project has been developed by the audio team of Multimedia Assistive Technology Laboratory (MATeLab) at the Università Politecnica delle Marche, which operates in the ambient assisted living context exploiting audio-visual domain features. This research is part of the HDOMO 2.0 project founded by the National Research Centre on Aging (INRCA) in partnership with the Government of the Marche region under the action "Smart Home for Active and Healthy Aging".

References

L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmüller, and P. Maragos. 2014. The dirha simulated corpus. In *Proc. of LREC*, volume 5.

S. Davis and P. Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Proc., IEEE Transactions on*, 28(4):357–366.

L. Deng. 2012. Three classes of deep learning architectures and their applications: A tutorial survey. *APSIPA Transactions on Signal and Information Processing*.

F. Eyben, F. Weninger, F. Gross, and B. Schuller. 2013. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proc. of the 21st ACM international conference on Multimedia*, pages 835–838. ACM.

A. Graves. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

H. Hermansky. 1990. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752.

D. J. Hermes. 1988. Measurement of pitch by subharmonic summation. *The journal of the acoustical society of America*, 83(1):257–264.

G. Hinton, S. Osindero, and Y. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

G. Hinton. 2010. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

T. Houtgast and H. J. M. Steeneken. 1985. A review of the mtf concept in room acoustics and its use for estimating speech intelligibility in auditoria. *The Journal of the Acoustical Society of America*, 77(3):1069–1077.

E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller. 2014. Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks. *Proc. of 39th IEEE ICASSP*.

E. Principi, S. Squartini, F. Piazza, D. Fuselli, and M. Bonifazi. 2013. A distributed system for recognizing home automation commands and distress calls in the italian language. In *Interspeech*, pages 2049–2053.

L. R. Rabiner and B. Juang. 1993. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs.

F. Weninger, J. Bergmann, and B. Schuller. 2014. Introducing CURRENNT – the Munich Open-Source CUDA RecurREnt Neural Network Toolkit. *Journal of Machine Learning Research*, 15.

M. Wolf and C. Nadeu. 2014. Channel selection measures for multi-microphone speech recognition. *Speech Communication*, 57:170–180.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. 1997. *The HTK book*, volume 2. Entropic Cambridge Research Laboratory Cambridge.