Doctoral Thesis

Lexical Resources as Semantic Layers on top of Language Models

Polo delle Scienze della Natura
Computer Science Department
University of Turin

Davide Colla

Turin, July 20th, 2022

Supervised by Prof. Daniele P. Radicioni

To speak a language is to take on a world, a culture.

— Frantz Fanon

Language is a process of free creation; its laws and principles are fixed, but the manner in which the principles of generation are used is free and infinitely varied. Even the interpretation and use of words involves a process of free creation.

— Noam Chomsky

Abstract

Lexical resources are central to Natural Language Processing (NLP). Over the years, lexical resources have been successfully employed to tackle many diverse NLP tasks, such as Word Sense Disambiguation, Question Answering, Information Retrieval, Sentiment Analysis and many others. In the same spirit of lexical resources, language models are a long-standing component of Natural Language Processing, they are central to many different tasks. In the last years, language models have attracted considerable attention, thus leading to a rapid evolution through neural architectures, ending up with impressive models such as BERT and GPT-2. Such a rapid evolution of language models plays a fundamental role in the development of lexical resources. In this work two lexical resources are presented, that reflect the evolution of neural architectures: LESSLEX and SE-MACAROON.

LESSLEX is a set of embeddings that extends the terminological embeddings of ConceptNet Numberbatch by building semantic representations that co-exist in the same semantic space with those acquired at the term level: for each term we have thus the 'blended' terminological vector along with those describing all senses associated to that term. LESSLEX has been extensively assessed on a wide variety of tasks, such as word similarity, conceptual similarity, semantic similarity, contextual similarity and semantic text similarity.

Focusing on the semantic similarity task —and on the metrics usually employed to compute semantic similarity— allowed us to make explicit the sense identification task: semantic similarity accounts for similarity ratings only, conversely sense identification involves identifying which senses which actually underlie that similarity score. We posit the sense identification as a natural and crucial complement to the semantic similarity task.

Additionally, we developed a second lexical resource, SE-MACAROON. SE-MACAROON contains vectorial representations built by integrating context sensitive descriptions from BERT and the structured semantic information from WordNet. Different from the existing contextualized sense embedding techniques, SE-MACAROON represents word senses as collections of word embeddings rather

than conflating all of its occurrences into a unique representation. Such feature provides the resource with a sort of lexical memory, storing the ideal representations of a word sense taken in context, and collecting sense representations possibly close to several contexts of usage. This allowed us to devise a novel approach to WSD, exploiting multiple occurrences for each word sense and obtaining results that directly compare to the state-of-the-art sense embeddings.

We eventually investigated the role of language models in a specific application setting, where we explored how suited perplexity is as a marker for measuring coherence in spoken language. We also investigated whether perplexity can be used as an automatic linguistic analysis tool to assist clinical diagnosis. The obtained results seem to corroborate our chief hypothesis: distributional sense-level representations allow to deal with lexicographic precision of semantic networks paired to the flexibility proper to distributional resources. Additionally, such lexical resources are built as a semantic layer sitting on top of language models: this yields as output symbolic information tied to the terminological space defined by language models representations. Such feature allows to compare terms and senses descriptions inspiring novel strategies to address a growing number of NLP tasks.

Acknowledgements

I would like to express my deepest gratitude to Professor Daniele Radicioni, who has followed and guided me during these years, assisting me in overcoming any doubt, giving me precious advice, his support and his encouragement. I would also like to thank him for his patience, without him none of these would have been possible. A very special thanks goes also to my colleagues Enrico Mensa and Matteo Delsanto: Enrico inspired and guided me through the obstacles, provided me with valuable advice and his support; Matteo supported me with tireless dedication and constant presence, sharing with me studies, joys and discouragement to which research can lead. I wish to express my deepest gratitude to Anna and Diego, they sheltered me during difficult times without any hesitation, guiding me and supporting me, also with their cheerfulness.

I'm extremely grateful to my family, for their support that allowed me to pursue this career. I wish to thank Alessandra for her care, patience and support, especially during the pandemic. To my brother Marco, who supported and encouraged me since I was born. I would like to express my gratitude to Enza and Nino for their support and immense care.

A special thank goes to all of my closest friends, Dino, Elisa, Federico, Gabriele, Giuseppe, Marco, Riccardo, Sabrina, Sara and Simone for their patience and unparalleled support.

Contents

List of Figures

List of Tables

1 Introduction

Lexical resources are central to Natural Language Processing (NLP). Over the years, lexical resources have been successfully employed to tackle many diverse NLP tasks such as Question Answering (Esposito, Damiano, Minutolo, De Pietro, & Fujita, 2020; Kaisser & Webber, 2007), Semantic Word Similarity (Jiang & Conrath, 1997; Speer, Chin, & Havasi, 2017), Relation Extraction (P. Li, Mao, Yang, & Li, 2019), Event Extraction (Ahn, 2006), Automatic Text Summarization (Pal & Saha, 2014; Tauchmann & Mieskes, 2020), Supersense Tagging (Ciaramita & Altun, 2006; Flekova & Gurevych, 2016) and Word Sense Disambiguation (WSD) (L. Huang, Sun, Qiu, & Huang, 2019; Levine et al., 2019; Navigli, 2009; Tripodi & Pelillo, 2017), thereby determining further impulse in developing newer lexical resources. The building rationale underlying such lexical resources may differ according to the considered task: more general lexical resources such as WordNet (G. A. Miller, 1995) may serve as a base to develop domain specific resources (J. McCrae et al., 2012) or to envision resources involving several different languages BabelNet (Navigli & Ponzetto, 2010).

Dealing with lexical resources involves to cope with word senses representation, that is, building representations for the different meanings related to a word. In the last decades the research took two parallel paths to represent word senses: symbolic approaches, resulting in resources such as WordNet, and resources following the distributional hypothesis (Harris, 1954). Symbolic information expects that the connection among lexical surface and meaning be made explicit, interfacing strings to concepts. Pioneering resources such as WordNet are significant undertakings which arrange symbolic information about word senses —that is, lexicographic information— in a machine-readable configuration, while adopting the overall structure of semantic networks. Such resources have been exploited as sense dictionary for many different NLP applications, such as text classifica-

tion (Scott & Matwin, 1998), word sense disambiguation (Lesk, 1986), computing semantic relatedness between pair of words (Patwardhan & Pedersen, 2006) and text summarization (El-Kassas, Salama, Rafea, & Mohamed, 2021).

Different from symbolic knowledge bases, distributional resources represent word senses through word embeddings: real valued vectorial representation for words in a multi-dimensional semantic space. Word embeddings have proven to be suitable for representing word meanings over a multidimensional Euclidean space, where distance acts like a proxy for similarity, and where similarity can be interpreted as a metric. Word embeddings have been successfully applied to a broad set of diverse NLP applications such as text classification (Lilleberg, Zhu, & Zhang, 2015), sentiment analysis (H. Liu, 2017), named entity recognition (Sienčnik, 2015) and word similarity (Speer et al., 2017). Thanks to the emergence of deep neural models for NLP, word embeddings started also to be exploited as initial weights for such models (Goldberg, 2017): neural models implicitly develop word embeddings when training for the language modeling task (Bengio, Ducharme, Vincent, & Janvin, 2003), thus pre-trained word embeddings may be exploited as initial word representation for newer language models.

The evolution of language models plays a fundamental role in lexical resources development as well as for many different NLP tasks, to the extent that newer NLP evaluation benchmarks such as Glue (Wang et al., 2018) and SuperGlue (Wang et al., 2019) need to be developed. Early language models estimate the probability only for a small group of words, thus resulting in sparse representations based on co-occurrences (C. Manning & Schutze, 1999). The introduction of neural networks boost the language models development allowing implementing textual memory together with dense representations (Devlin et al., 2018; Elman, 1990; Hochreiter & Schmidhuber, 1997; Radford et al., 2019). Word embeddings produced by neural models evolved according to the language model paradigm: whereas early neural language models built fixed word embeddings, recently proposed context sensitive representations have received considerable attention due to their versatility.

The focus of this work is the development of two lexical resources as semantic layer on top of language models. The first proposed resource is LESSLEX, a set of embeddings containing descriptions for senses rather than for words: while

word embeddings typically describe terms, LESSLEX contains different distributional representations for different word senses from a semantic network's vocabulary. For example, given the WordNet entry for *bank* intended as "sloping land (especially the slope beside a body of water)", vectorial representations for *riverbank, riverside, bank, coast* (i.e. all the synonyms of such meaning) are collected and combined to build the representation related to such word sense. Additionally, representing word senses allow dealing with multiple languages with a single characterization: word embeddings are strictly linked to the language of the corpus, by dealing with senses only a single vector for *financial institution* sense is needed, regardless of the language. Since dealing with multiple languages with the same representation is to date one of the chief challenges in lexical semantics, we decided to address diverse languages exploiting the multilingual nature of BabelNet and ConceptNet Numberbatch (Speer & Chin, 2016) thus generating distributional vectors for BabelNet's entries. LESSLEX vectors have been tested in a widely varied experimental setting, comprising semantic similarity for word in context and textual similarity, providing performances at least on par with state-of-the-art embeddings, and sometimes substantially improving on these.

Additionally, we tested LESSLEX vectors on semantic similarity task: the semantic similarity task consists in computing a score to assess the proximity of the meaning of two lexical items; in order to assess its accuracy, the computed semantic similarity score is customarily compared against similarity ratings provided by human annotators. Semantic similarity is a long-standing topic of investigation (see, e.g., works by Baddeley (1966a, 1966b); Schaeffer and Wallace (1969)), and in the last few years it has emerged as a central one: historically, this phenomenon is related to various aspects, such as the growing needs for elaborating natural language at large, and the wide availability of high quality word embeddings.

Assessing LESSLEX embeddings on semantic similarity task, allowed us making a complementary task explicit: the *sense identification task*. Basically, this task allows answering a —previously unseen, though— basic question: Which senses actually lie at the base of the similarity rating? In this setting we explored whether the semantic similarity task can be paired to such complementary task aimed at identifying which senses are actually involved in the semantic similarity rating.

In other words, we posit that sense identification is a natural and crucial complement to the semantic similarity. Addressing the sense identification task involves redesigning classical similarity metrics taking into account word senses responsible for the similarity rating: metrics for the semantic similarity task are designed to compute the similarity score only, sense identification task also requires explicating the senses underlying the similarity rating. Therefore, we defined novel semantic similarity metrics that favorably compare to the familiar cosine similarity maximization, both in the semantic similarity task, and in the sense individuation task: our novel metrics can be simply plugged into existing systems to replace the maximization strategy.

The second proposed resource is SE-MACAROON, a set of sense embeddings constructed on context sensitive representations from BERT. Different from LESSLEX, SE-MACAROON relies on contextualized word embeddings computed through BERT on a sense tagged corpus. Contextualized word embeddings have been employed effectively across several tasks in Natural Language Processing, as they have proved to carry useful semantic information. Despite their nature, context embeddings are still difficult to link to a structured knowledge base such as WordNet. The proposed approach builds sense embeddings extracting context sensitive representation exploiting BERT language model on a sense labeled corpus. Different from state-of-the-art contextualized sense embeddings, such as ARES (Scarlini, Pasini, & Navigli, 2020b), that build unique fixed representation for each word sense, our hypothesis is that word senses might be characterized more precisely through a set of vectors representing the occurrences of the word sense in ideal contexts. Starting from SemCor (G. A. Miller, Chodorow, Landes, Leacock, & Thomas, 1994), the largest, manually curated, sense labeled corpus, we collected all the sentences in which the word w occurs intended as the sense s, and built a set of context sensitive embeddings through BERT. We assessed SE-MACAROON embeddings in Word Sense Disambiguation (WSD) task, providing performances comparable with state-of-the-art embeddings. We subsequently assessed the impact of the proposed WSD approach, providing figures supporting our hypothesis.

Over the years, language models have been widely employed across several NLP tasks, such as information retrieval (Berger & Lafferty, 1999; Hiemstra, 2001;

D. R. H. Miller, Leek, & Schwartz, 1999), offensive language detection (Colla, Caselli, Basile, Mitrović, & Granitzer, 2020; Xu, Liu, Shu, & Yu, 2019), sentiment analysis (Singh, Jakhar, & Pandey, 2021), spelling error correction (Ivanov, Musa, & Dulamragchaa, 2021), sentence classification (Ali Awan et al., 2021) and many others. Despite the success language models have received, no general consensus on the evaluation framework has been reached: the assessment may be cast to more extrinsic setting, thus assessing the model on a higher-level task, or to a more intrinsic fashion by exploiting the perplexity metrics (Goldberg, 2017). Paired with language models evolution, the analysis of human language has recently emerged as a research field that may be helpful to analyze for diagnosing and treating mental illnesses. In fact, in the last decade NLP techniques have become a common tool to support research on psychotic disorders (Fritsch, Wankerl, & Nöth, 2019). Following this research line, perplexity has been recently proposed as an indicator of cognitive deterioration (Frankenberg et al., 2019).

The final section of this thesis provides an exploration on this issue: Whether the perplexity metrics can be interpreted as a semantic coherence marker, thereby allowing us to employ language models in the early detection of psychotic disorders. After having presented two resources, in this chapter we show how to employ an intrinsic metrics (originally concerned with evaluating the 'fit' of a language model to actual language) to predict the insurgence of a broad class of cognitive impairments, as these affect linguistic production. We tested whether the perplexity computed employing a language model acquired based on speeches from healthy subjects can be useful in discriminating healthy subjects from people suffering from mental disorders. In this respect, we compared perplexity ratings obtained through Bigrams and GPT-2 on a tripartite experiment: we first examined whether perplexity can be deemed as reliable to analyze speech transcripts under an intra-subject and discourse-level coherence perspective; we then assessed it by examining different subjects, and compare perplexity scores as computed through LM built by employing different architectures; finally, we tested perplexity to discriminate healthy subjects from subjects affected from Alzheimer Disease. The results seem to support our hypothesis proving the coherence of the perplexity metrics, especially on language-specific trained models. Furthermore, we show that such language mod-

els, after accurate pre-training process, together with the perplexity metrics might be exploited as additional feature accounting for decision process.

LESSLEX and SE-MACAROON rely on different language models grasping similar knowledge: the former supplies distributional and opaque knowledge which can be exploited to establish distances both at the terminological and conceptual level. Such knowledge is built upon static word representations comparable to those provided by early language models. The latter one provides conceptual representations that can be used to deeply analyze and understand senses underlying words in textual data. Such representations are built by employing context sensitive language models, thus obtaining different nuances of the same word sense according to the lexicalization and the sentence in which it occurs. Although different in nature, both LESSLEX and SE-MACAROON build a semantic layer on top of language models: representations supplied by these resources live in the same semantic space, defined by their common grounding on WordNet. This feature is deemed as helpful in contributing to the interpretability and transparency of such models.

In what follows a compact list of the main research questions addressed, along with the contributions of the thesis is provided.

- The chief research question addressed in the first part of the work is as follows: how symbolic knowledge can be integrated with distributional resources to build sense embeddings?

  LESSLEX sense embeddings were introduced. LESSLEX implements a novel approach aimed at building static vector representation for senses as an integration between symbolic and distributional knowledge. Additionally, LESSLEX sense embeddings share the same semantic space defined by the adopted distributional resource(Chapter 4)[1]. LESSLEX was experimentally proved as a robust, wide-coverage resource. It was shown that the purely conceptual representation delivered by LESSLEX allows to deal with multi- and cross-lingual application settings with no need for retraining.

- The second research question addressed in this thesis is the following: How

[1]The work illustrated in Chapter 4 was published in: Davide Colla, Enrico Mensa and Daniele P. Radicioni. LESSLEX: Linking multilingual Embeddings to SenSe representations of Lexical items *Computational Linguistics*, 46(2), pages 289-333, June 2020.

to exploit shared conceptual and terminological spaces to build novel and more accurate metrics for semantic similarity? Is it possible to envisage a task closely related to semantic similarity, which is focused on identifying which senses are actually involved in the semantic similarity rating?

A novel task —the Sense Identification task— was individuated, underlying and complementing the semantic similarity task. To these ends, an annotated corpus was released, the Sense Identification Dataset (Chapter 5).[2] Two novel metrics were proposed, $\mathcal{R}$-sim and $\mathcal{N}$-sim, aimed at addressing both semantic similarity and sense identification; such metrics were observed to obtain more accurate results than the popular cosine similarity maximization; in the experimentation a pool of state-of-the-art sense embeddings was employed.

- The third research question addressed in this thesis is: Can multiple contextual vector descriptions be helpful in representing senses? Is it possible to exploit such representations to deal with the Word Sense Disambiguation task?

  A novel approach for constructing contextual sense embeddings was introduced, characterizing each word sense through multiple vector descriptions (Chapter 6).[3] SE-MACAROON sense embeddings were built by conflating WordNet and the BERT language model. A novel Word Sense Disambiguation strategy was proposed; the multiple vectorial descriptions underlying the SE-MACAROON embeddings improve accuracy over the more popular average vector representations.

- The fourth research question addressed is whether language models can be used to analyse linguistic deficits featuring the speech of cognitively impaired subjects, and to assist specialists in the early diagnosis of specific disturbances afflicting linguistic production, such as Alzheimer Disease.

  A real-world application scenario involving language models was consid-

[2] The work illustrated in Chapter 5 was published in: Davide Colla, Enrico Mensa and Daniele P. Radicioni. Novel metrics for computing semantic similarity with sense embeddings, *Knowledge-based systems*, 206:106346, 2020; and in Davide Colla, Enrico Mensa, and Daniele P. Radicioni. Sense identification data: a dataset for lexical semantics. *Data in Brief*, page 106267, 2020.

[3] The work presented in Chapter 6 has not been previously published. An extended version of the research therein is under preparation.

ered, where the perplexity metrics was employed as a semantic coherence marker. In this study, evidence was provided that language models can be employed to effectively assist specialists in the early detection of mental disturbances and psychotic disorders (Chapter 7).[4]

[4] A revised version of the work presented in Chapter 7 has been submitted to the Artificial Intelligence in Medicine Journal, and successfully passed through two rounds of review.

2 Preliminaries

This work exploits the interaction between language models, vector-based representations and semantic representations. In this Chapter we provide the essential background to such topics. The Chapter is organized as follows: first we introduce language models (Section 2.1), where we report on both N-grams (Section 2.1.1) and neural language models together with the main traits of the neural architectures actually employed to acquire such models (Section 2.1.2); the second part of the Chapter is focused on lexical resources (Section 2.2) illustrating distributional resources (Section 2.2.1) and semantic networks (Section 2.2.2).

2.1 Language Models

Language models are statistical inference tools that allows estimating the probability of a word sequence $W = \{w_1, \ldots, w_k\}$ (Goldberg, 2017; C. Manning & Schutze, 1999). For example, a language model is able to assign the probability to the sentence *I shot an elephant in my pajamas*. More frequently, language models are employed to compute the probability of seeing a given word following a sequence of words —usually called context—. For example, what is the probability of seeing the word *pajamas* after the sequence *I shot an elephant in my*? The probabilities assigned by language models are the result of a learning process, i.e. the training phase, in which the model is exposed to a particular kind of textual data —i.e. the training corpus—. The goal of the training process is to teach the model to predict sentences that closely resemble the sentences seen during learning.

Formally, the Language Modeling task is defined as the assignment of probability to any possible sequence of words $W = \{w_1 \ldots w_k\}$, so as to compute

$p(W)$ (Goldberg, 2017). Such probability can be computed as

$$p(W) = \prod_{i=1}^{k} p(w_i|w_1, \ldots, w_{i-1}), \tag{2.1}$$

where the probability of each word is conditioned on the preceding context. Depending on the adopted language model as well as the assumptions on the conditioning factor the probability of the sentence may be framed differently.

2.1.1 N-grams

The simplest idea is to consider words individually, that is, each word is a single unit, this is what is called the unigram model. A unigram model considers no preceding context, given the sequence of words $W = \{w_1 \ldots w_k\}$, the probability of the sequence is defined as:

$$p(W) = \prod_{i=1}^{k} p(w_i),$$

where the probability of the *i*-th word $p(w_i)$ can be estimated by exploiting the frequency of the word w_i in the training corpus. The natural extensions of unigram models, are the *N*-gram models, where N is an integer indicating the size of the context, i.e. the preceding $N-1$ words are exploited to estimate the probability of the *N*-th term of the sequence (Jurafsky & Martin, 2014). Formally, in the *N*-gram setting, the probability of the sequence W is defined as:

$$p(W) \approx \prod_{i=1}^{k} p(w_i|w_{i-N+1}, w_{i-N+2}, \ldots, w_{i-1}), \tag{2.2}$$

where the probability of each word is conditioned on the preceding context. In this case only blocks of few (exactly N) words are considered to predict the whole W: we can thus predict the word sequence based on N-grams, that are blocks of two, three or four preceding elements (bi-grams, tri-grams, four-grams, respectively). Since the *N*-gram models are statistical models, their knowledge strictly depends on the training corpus, that is, the probability distribution reflects the language property of the training corpus itself. The natural consequence of estimating the probability of the upcoming word, relying on the preceding $N-1$ words, is that *N*-

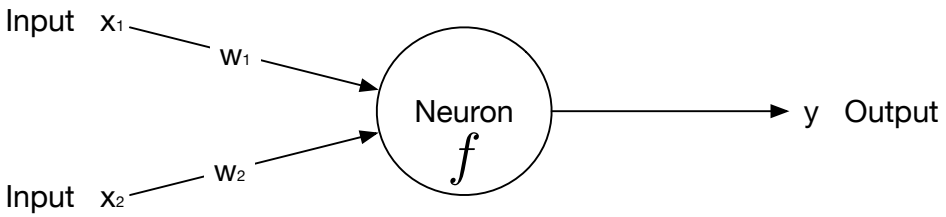

Figure 2.1: Typical neuron architecture. The input data are represented as a sequence of real valued features x_1, x_2, the neuron applies the activation function f to the weighted sum of input features and produces the output y.

gram models obtain better performances at representing the training corpus as N increases, with the drawback of making the estimation of $p(w_k|w_1 \dots w_{k-1})$ harder. Increasing the context size, also, involves dealing with data sparsity drawback: the larger the context the less likely it is to find more than one sequence with the same length in the training corpus. That is, the probability for all the N-grams that does not occur in the training corpus has to be estimated, this gives rise to plenty of sequences with the same low probability and few sequences with high probability. In order to deal with N-grams not occurring in the training corpus, called out-of-vocabulary N-grams, language models have been provided with an additional step of regularization, to allow a non-zero probability to unseen N-grams (Gale & Church, 1994; Kneser & Ney, 1995).

2.1.2 Neural Networks and Neural Language Models

Neural networks are computational systems inspired by the human brain. The history of neural networks started with the McCulloch-Pitt neuron —or unit—, that is, a computational model of human neurons, entirely described with propositional logic (McCulloch & Pitts, 1943). Modern neural networks organize neurons into layers, each unit is connected to each unit of the subsequent layer through *synapses* or *edges*. Each layer of the network accepts as input the output of the preceding layer, performs some transformation on the received data, and produces an output according to the layer architecture. Different layers apply different transformations; edges, in turn, are usually provided with a weight, an integer expressing the strength of the connection among two neurons, which is usually exploited to alter data coming from the preceding layer. A graphical representation of a simple neuron is depicted in Figure 2.1. A single unit receives as input real valued features,

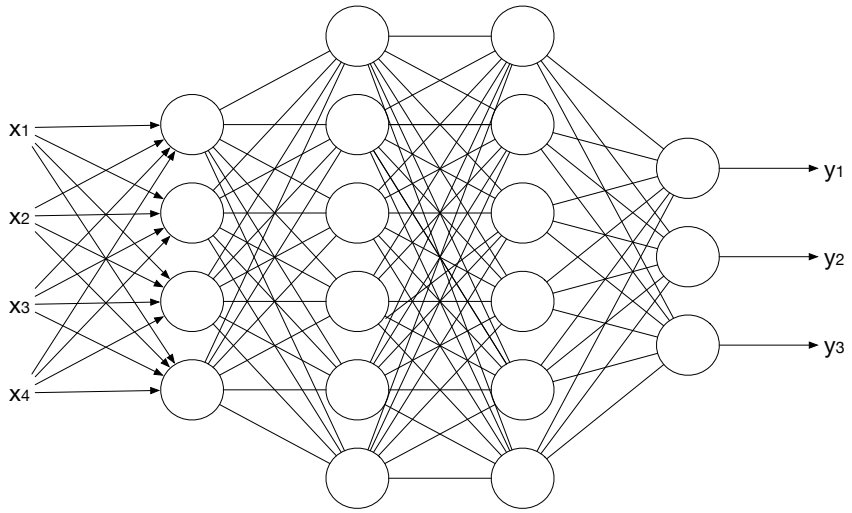

Figure 2.2: Multi-layer neural architecture. The input layer takes the input data $x_1, \ldots, x_4$ and passes the information to the first hidden layer. Hidden layers are stacked and connected to each other to obtain a higher level of abstraction on data. The last hidden layer passes the information to the output layer which computes the final representation $y_1, \ldots, y_3$ with a different activation function.

$x_1, x_2, \ldots, x_n$, which represent the input data, then, it combines the input features through a weighted sum:

$$y = b + \sum_{i=1}^{n} w_i x_i$$

where w_i are the weight of the edges connecting data with the input layer and b is a bias term which is an additional term that allows shifting the data transformation by adding a constant. Additionally, each neural unit is then provided with an activation function, that is, a non-linear function applied to the weighted sum of input features and bias term. So the final output value y of a neural unit is:

$$y = f(b + \sum_{i=1}^{n} w_i x_i)$$

where f is the activation function of the single neuron. Different activation functions correspond to different transformation, and several functions may be employed depending on the addressed task.[1] Neurons are organized into layers, and each layer may play a different role in the architecture, also according to the activation function. An example of a multi-layer neural network is depicted in Figure 2.2. In particular, the layers can be partitioned in three macro-categories according to

[1]The most popular activation functions are: sigmoid function, Rectified Linear Unit (ReLU), the hyperbolic tangent (tanh) and the SoftMax function.

their position: (i) the input layer, which is dedicated to take input data, represented as numerical features, and to pass information to the hidden layer; (ii) hidden layer, made of non exposed units, is the heart of neural computation, all the major transformations happen here, and all neural units of the layer share the same activation function; (iii) output layer, is the final layer of the architecture, usually apply a different function, with respect to hidden layers, so to build the final representation.

The role played by hidden layers is fundamental in computing the final representation. Each hidden layer provides additional abstraction to the neural model: in fact, hidden layers can be stacked one on each other to obtain a higher abstraction level. Architectures provided with multiple hidden layers are usually called deep neural networks. Since neural networks deal with real valued representations of data, it is essential to extract features from data, which are texts in our case, and map them to a numerical vector representation —usually called embedding— able to fully grasp the key features from the input data. The role of vector representations is central to neural models: modern neural networks are provided with an embedding layer, which is responsible for the creation of a fixed-length vector for each element of the input sequence. It is worth noting that these vector representations mitigate the data sparsity problem by building a continuous space, each word has its corresponding vector in the network space. Since language models deal with textual data, we report on the most popular neural networks for Natural Language Processing (NLP).

In general, dealing with sentences involves dealing with ordered sequences of words: in order to fully seize the meaning of a sentence, a model should be able to account for word ordering information. Given the relevance of modelling the word order in sentences, one of the most employed architectures is the Recurrent Neural Network (RNN). RNNs have been largely employed in text processing due their ability of processing input data as sequences: that is, RNNs are able to grasp and model word order (Elman, 1990). RNNs are particularly suited to process sequence data thanks to the internal loop they are provided with, that is, the input of each unit is conditioned by the output of its own output at the preceding iteration. Figure 2.3 shows a graphical illustration of a single RNN unit. The hidden state encodes a memory for the context, it provides all the additional information that is

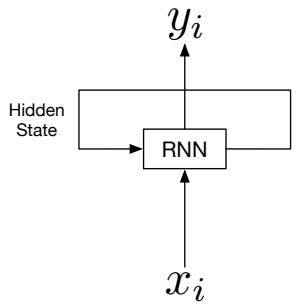

Figure 2.3: RNN unit: takes x_i as input, and computes the output representation y_i as a composition of x_i and the hidden state of the preceding time step.

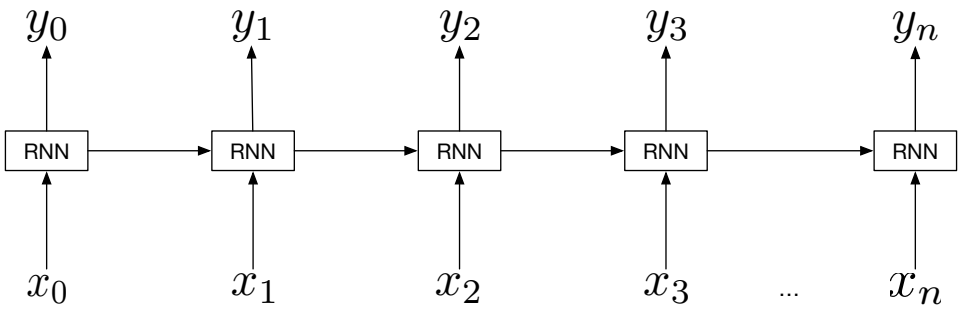

Figure 2.4: Representation of an RNN unrolled.

exploited to compute the output of later time steps. An RNN model does not fix a limit for the length of the input sequence, however, for a finite input sequence, an RNN model can be represented as unrolled, a graphical illustration is shown in Figure 2.4. In the unrolled setting we can see that the last hidden state depends on the entire input sequence, in that the prediction of the next word is conditioned on the previous words in the sentence. The ability of conditioning the prediction of the next word to the preceding context, that is, dealing with sequences, is the most appealing feature of the RNN architectures, nevertheless, these models struggle in modeling the context when facing long range dependencies. More precisely, some tasks can be addressed by accounting for the recent information only, for example the five words before the next one. Conversely, to tackle other tasks, we need the information from the whole sequence, for example, let us consider the sentence *I was born in Italy and then, to follow my work, I moved to Germany. However, my family still lives in Italy.*, to predict the last word *Italy* we should exploit the information at the beginning of the sentence, that is, the distance between the prediction and the useful context is quite large. Unfortunately, however, RNNs are progressively less suited to model dependencies as the intervening distance between dependents

grows (Bengio, Simard, & Frasconi, 1994).

Given the difficulties in seizing long range dependencies, RNNs were quickly replaced by the Long Short-Term Memory networks (LSTM) (Hochreiter & Schmidhuber, 1997). LSTMs are RNNs specifically devised to learn long range dependencies; this is obtained by providing units with an explicit context memory that conveys the information about the preceding context through time. The context representation is performed through two main operations: (i) forgetting information no longer needed from the context and (ii) adding new information probably needed for next word prediction. Both sub-tasks are addressed through specialized neural units called gates, that manage the information flow through the memory state and the output of the LSTM cell. These gates follow the same design pattern: a neural layer followed by a sigmoid activation; the output is then combined through multiplication with the information to be filtered. The sigmoid function, combined with point-wise multiplication, allows the gate to decide whether to retain information (that is by setting the output to 1) or to forget the information, setting the output to 0. More precisely, an LSTM unit is composed by three gates: the forget gate, which is responsible for deleting information no longer needed from the context representation; the input gate, concerned with adding new information to the context representation; the output gate, that is aimed at computing the output representation, which coincides with the unit hidden state, by accounting for the preceding hidden state and the new context representation. A graphical illustration of an LSTM unit is provided in Figure 2.5.

LSTMs are particularly suited to deal with sequences and long range dependencies. Despite their abilities, simple LSTM models can only work with fixed length input sequences, but some tasks such as machine translation or speech recognition are likely better expressed by dealing with sequences whose length is not fixed. The Sequence-to-Sequence (S2S) model has been proposed in 2014 (Sutskever, Vinyals, & Le, 2014) to overcome such limitations. The S2S model relies on LSTMs to map a sequence $x_1, \ldots, x_n$ of an arbitrary length to another sequence $y_1, \ldots, y_k$ where k may be different from n. In this setting, the input sequence is processed by an encoder, which compresses the sequence to a fixed length vector representation C. The decoder is then initialized on the C vector, and predicts the output token by

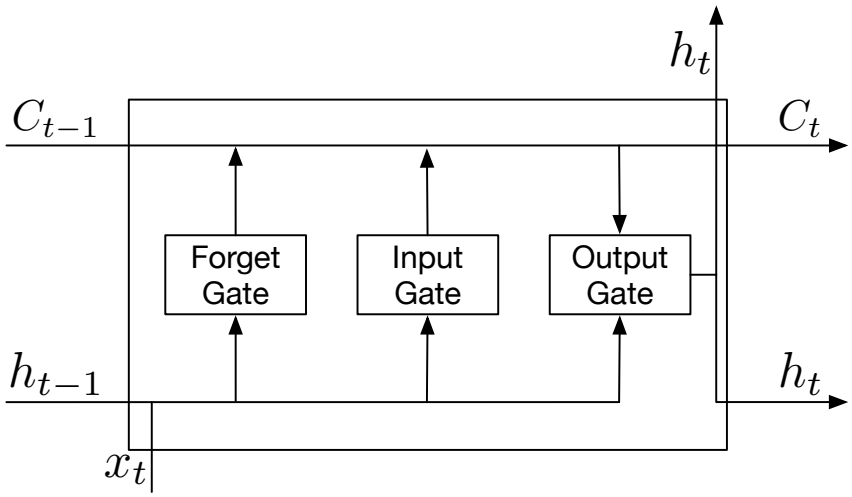

Figure 2.5: Representation of an LSTM unit. Here, C_{t-1} and h_{t-1} are the context representation and the hidden state coming from the preceding unit, respectively. The input token is represented by x_t. The output of the cell corresponds to its hidden state at the current time step h_t. The updated representation of the context C_t and the hidden state h_t are then forwarded to the next LSTM unit.

token, accounting for the previously predicted token at each time step. A graphical representation of the S2S architecture is depicted in Figure 2.6.

Despite the ability of LSTM architectures to deal with long range dependencies, these models still struggle in representing larger pieces of text and suffer from high training time due to the recurrent connections which build these units. Additionally, the S2S architecture suffers from the loss of informative load in compressing the whole input sequence into a single fixed length vector representation. Transformers (Vaswani et al., 2017), together with the attention mechanism (Bahdanau, Cho, & Bengio, 2014) alleviate these problems by both increasing the amount of exploited information from the context, and getting rid of the recurrent connections. The attention mechanism has been designed to alleviate the difficulties in S2S models; this is done by allowing the decoder to directly exploit the encoder's hidden states rather than just using the final context representation provided by the encoder itself. Adopting an attention mechanism allows the model to selectively focus on parts of the input that are likely the most useful. The attention mechanism is particularly suited to address tasks which need to take decisions relying on parts of the input data. An illustration of the attention mechanism fitted in the S2S setting is depicted in Figure 2.7. Attention plays a key role in the Transformer architecture, in particular, the model illustrated in Figure 2.7 follows the S2S design pattern

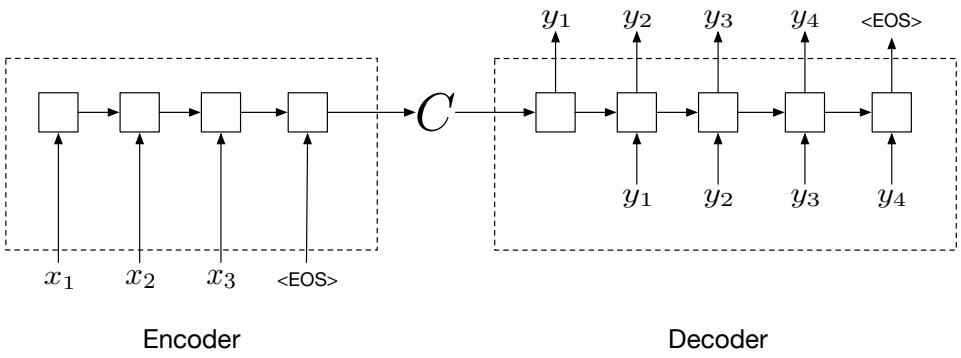

Figure 2.6: Representation of an S2S setting. Here <EOS> represents the end of the sentence. The input sequence $x1, x2, x3$ is processed by the encoder and compressed to the context vector representation C. The context vector is then forwarded to the decoder which predicts the output sequence y_1, y_2, y_3, y_4 by taking as input the previously predicted token at each time step.

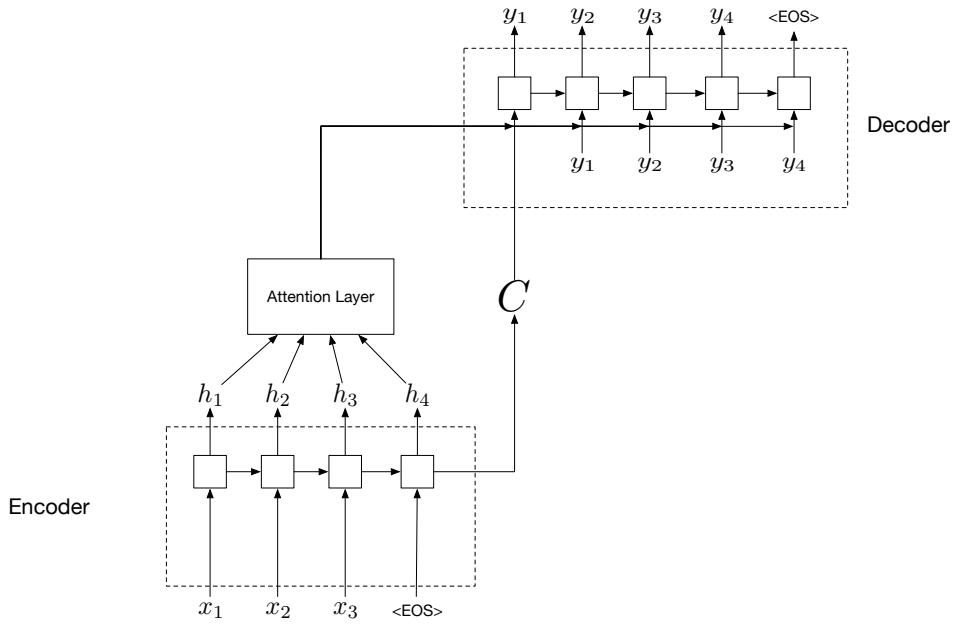

Figure 2.7: Attention mechanism in an S2S setting. Here each prediction of the decoder relies on both the previously predicted token and a composition of the encoder hidden states.

where the encoder processes the input sequence, the output is then forwarded to the decoder which curates the output predictions. In this setting, we will refer to the encoder-decoder model as the Transformer block. Since Transformers get rid of recurrent connections, which allow models to deal with sequences, the encoder represents the input through a combination of word representations and information about the position of words in the input sentence. In so doing, the model is able to account for ordering information. After this first operation, the encoder block is made of an attention layer followed by a simple neural layer which is deputed to compute the context representation. As for encoder, the decoder block combines the previously predicted word representations with the positional information to keep track of the order of the words, and forwards these vectors through an attention layer that selects the most useful information among the predictions. After these first steps the decoder combines the information from previously predicted tokens with the context representation, from the encoder, through another attention layer, and finally a simple neural layer is concerned with computing the output representation. Most popular models consist of several Transformer blocks stacked one on each other, this allows the model to increase its abstraction capabilities, the more the number of stacked blocks the more abstract the representation that can be calculated. A graphical illustration of a transformer block is provided in Figure 2.8.

Neural language models (NLMs) are language models based on neural networks. Such models improve the language modeling capabilities of n-grams by exploiting the ability of neural networks to deal with longer histories. Additionally, neural models do not need regularization steps for unseen n-grams and address the data sparsity curse of n-grams by dealing with distributed representation. The predictive power of neural language models is higher than n-gram language models given the same training set. Despite the great improvement of neural language models on NLP tasks, these models are affected by higher training time rather than n-gram language models. Since the introduction of Transformers, such models have been widely adopted and improved to address diverse Natural Language Understanding benchmarks such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019). With the introduction of highly-scalable Transformer

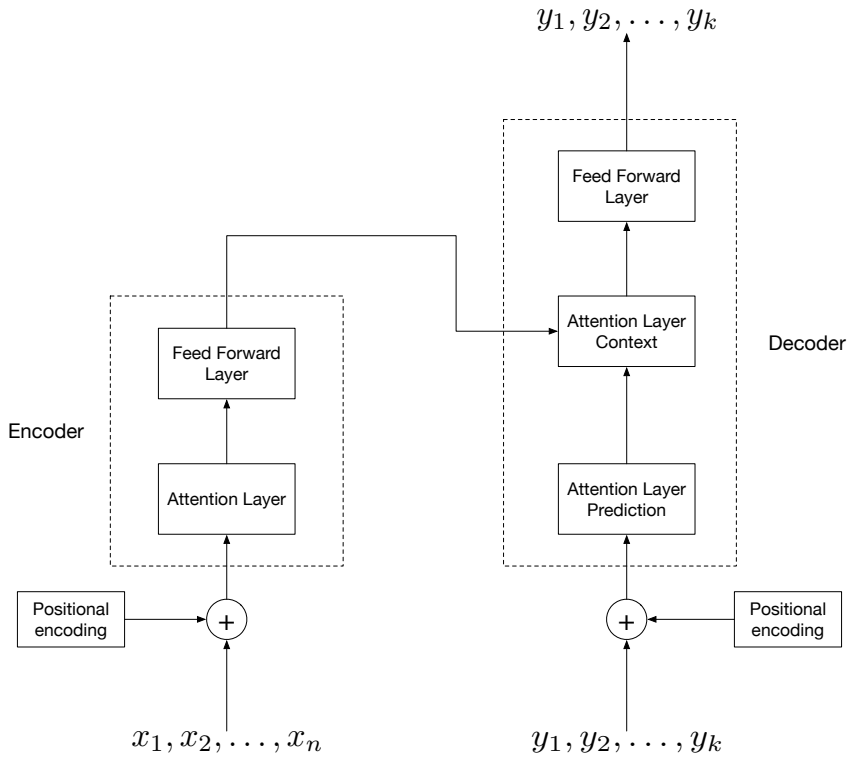

Figure 2.8: High level representation of transformer block. The input sequence $x_1, x_2, \ldots x_n$ is combined with positional information to account for ordering properties. The input is then processed by the attention layer of the Encoder and a simple neural layer aimed at representing the whole input sentence. The Encoder output is then combined with previously predicted tokens from the Decoder through another attention layer, and then, the last layer computes the output representation for each input token.

architectures two kinds of very deep NLMs emerged: causal (or left-to-right) models, primarily represented by the Generative Pre-trained Transformer (Radford et al., 2019) where the objective is to predict the next word given a past sequence of words; and masked models, where the objective is to predict a masked (i.e., hidden) word given its surrounding words, of which the most prominent example is the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). The difference in training objectives results in these two varieties of NLMs specializing at different tasks, with causal models excelling at language generation and masked models at language understanding. Models such as BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019) started a whole host of experiments and triggered an intense research activity to improve such models (Brown et al., 2020; Lan et al., 2019; Y. Liu et al., 2019; Raffel et al., 2019; Yang et al., 2019). Given the relevance of the two mentioned models, we briefly report on both BERT and GPT-2 as precursors to more recent lines of NLP research as well as responsible for main advances in addressing NLP benchmarks.

BERT

BERT is a large language model based on Transformers, but it differs in the training objective: the key innovation is applying the bidirectional training of Transformers to language modeling. Using BERT involves to deal with two different rather important phases: *pre-training*, where the model is exposed to unlabeled textual data so as to learn the main features from the type of employed texts; and *fine-tuning*, where the pre-trained model is fine-tuned to address a specific task by exploiting labeled data. During the pre-training phase BERT is designed to address two unsupervised predictive tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The MLM is what gives BERT the bidirectional attribute: in this setting, a small percentage of the input tokens —15% in the released models— is masked and the training objective is to predict those masked tokens. For example, the sentence *I shot an elephant in my pajamas* may be rewritten into *I shot an [MASK] in my pajamas* and the BERT target is to predict the word *elephant* istead of the *[MASK]* token. Unlike causal language models pre-training, the MLM objective allows representing both left and right context thus making the training bidirec-

tional at the cost of misaligning the pre-training with the fine-tuning: the *[MASK]* token does not appear during fine-tuning. To mitigate the misalignment issue, the pre-training process does not always replace the masked word with the *[MASK]* token; in some cases (20%), the word is replaced with another word randomly taken from the dictionary or just left unaltered the word itself. The loss function is computed by accounting for the masked tokens only, thus ignoring the prediction on non-masked words. The NSP training objective has been devised to allow BERT to learn the relationship between sentences. The NSP involves selecting pairs of sentences A and B: in half of the cases the sentence B directly follows the sentence A, while in the remaining half the sentence B is randomly selected within a textual corpus. The purpose of the training objective is to learn whether sentence B directly follows A or not. Since many NLP tasks involve learning the relationship between sentences, such as, for example, Recognizing Textual Entailment (RTE) or Question Answering (QA), the NSP objective is to strengthen the model by precisely learning the relationship between pairs of sentences.

The BERT architecture follows the Transformer's design pattern reported in Figure 2.8; the difference is that BERT is made of stacked Transformers blocks, one on another, to increase the abstraction level. Stacking Transformers blocks not only allows dealing with more and more abstract representations, but also to reproduce the NLP pipeline, that is, different BERT layers deal with different linguistic levels (Tenney, Das, & Pavlick, 2019; Tenney, Xia, et al., 2019). In Figure 2.9 we report a graphical illustration of the BERT model's input as provided by the authors (Devlin et al., 2018). Given that BERT has to deal with pair of sentences as well as masked tokens, the authors, provided the model with two special tokens, the *[CLS]* token is placed at the beginning of the first sentence and is used from following classification layers placed on top of BERT, while the token *[SEP]* is used as separator, to mark the end of a sentence. The sentence embeddings reported in the figure are used to distinguish between the sentence A and B for the NSP task, while the positional information is accounted through the positional embeddings.

Once the pre-training of the model has been completed, the model may be fine-tuned to address a specific NLP task. For each downstream NLP task a fine-tuning process is needed, in particular, to specialize a pre-trained model is sufficient to

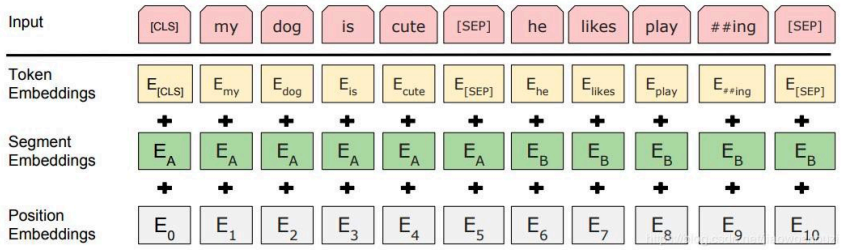

Figure 2.9: Representation of the BERT model's input. Token embeddings represent the word vectors, positional embeddings represent encode the ordering information of words in sentences. The segment embeddings are used to distinguish between the two input sentences. The [SEP] special token is used to mark the end of a sentence, as separator, while the token [CLS] can be used for classification purposes. Figure taken from (Devlin et al., 2018).

plug a classification layer on top of BERT relying on the encoded representation for words as well as for the [CLS] token. For example, a sentence pair classification task such as QA or RTE, a simple classification layer may be plugged on top of the Transformer's output relying on the [CLS] token. We refer the readers to Devlin et al. (2018) for an exhaustive list of design pattern for different kind of task.

GPT-2

GPT-2 is a large language model based on Transformers and trained to predict the next word given the preceding context (Radford et al., 2019). GPT-2, like traditional language models, predicts one token at a time, and the new prediction is appended to the input sequence for next time step. Inspired by P. J. Liu et al. (2018), which proposed the Transformer-Decoder architecture, GPT-2 is made of stacked decoder blocks only. More precisely, the Transformer-Decoder block is very similar to the decoder of the Transformer architecture; it simply gets rid of the encoder block as the contextual attention layer of the decoder. The GPT-2 architecture is portrayed in Figure 2.10.

The GPT-2 model has been trained 40GB of Internet text carefully selected for quality, that is a selection of documents curated or approved by humans. One main trait featuring the training data selection is that many different domains have been exploited as data sources; this allows the neural network to model language properties from each such domain, avoiding a strong polarization for a single one. Additionally, it is worth noting that the number of stacked decoder blocks impacts on performances, as the number of levels increases the language modeling capabilities

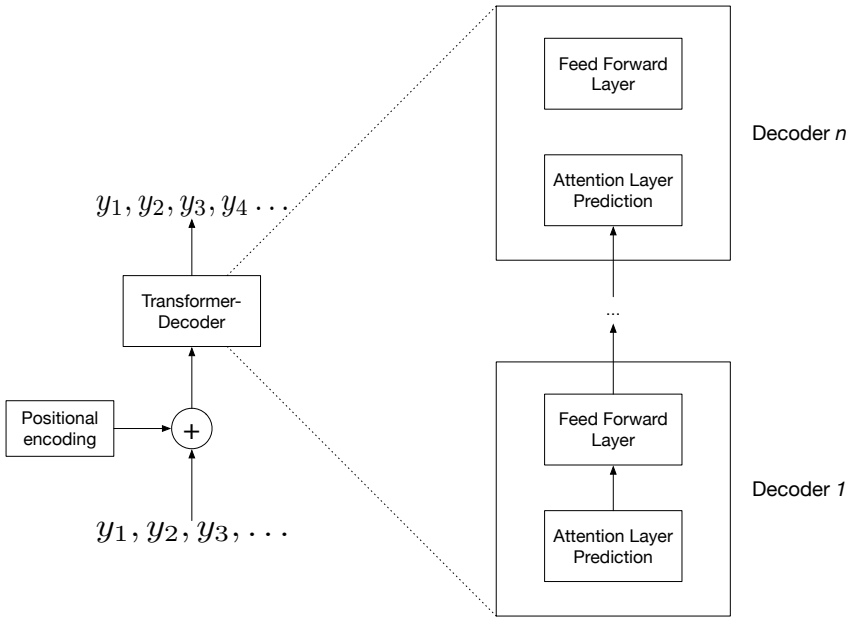

Figure 2.10: Representation of GPT-2 architecture. The model is made of n stacked decoder blocks. The input sequence $y_1, y_2, y_3, \ldots$ is processed by the n stacked encoders that form the Transformer-Decoder block. The last decoder produces the next token y_4 that is appended to the input sequence for the next time step. Each decoder is composed of an attention layer dedicated to processing the input sequence, and a simple neural layer that computes the output representation.

improve.

2.2 Lexical Resources

Since the purpose of this work is to build lexical resources on top of language models, we will briefly report on two families of lexical resources. In particular we are interested in lexical resources aimed at representing word senses according to two different principles: distributional resources, that represent word senses through dense vectorial representations, and semantic networks that represent word senses as vertices of a labeled graph.

2.2.1 Distributional resources

Pioneering works in the vector semantics area postulated that the meaning of a word always related to the context in which it occurs, depending on its usage in a language (Firth, 1935). This kind of meaning representation is known as the Distributional Hypothesis (Harris, 1954). The distributional hypothesis states that words

that occur in similar contexts tend to convey similar meanings, for example if the word w_i and the word w_k often occur in the same context, then they probably have close meanings; if they are interchangeable in the same contexts of occurrence, then they are synonyms. For example, in the sentences 'We used the *board* to shut down the power plant' and 'We used the *panel* to shut down the power plant', the words *board* and *panel* are intended with the same meaning.

Several techniques have been devised to acquire the distributional profiles of terms, usually in the form of dense unit vectors of real numbers over a continuous, high-dimensional Euclidean space. In this setting each word can be described through a vector, usually called *word embedding*, and each such vector can be mapped onto a multidimensional space where distance (such as, e.g., the Euclidean distance between vectors) acts like a proxy for similarity, and similarity can be interpreted as a metric. As a result, words with similar semantic content are expected to be closer than words semantically dissimilar. Early works in this family started by generating vectors from co-occurrence matrices (Harman, 1993; Schütze & Pedersen, 1997), optionally treated with latent semantic indexing (Landauer, Foltz, & Laham, 1998), or point-wise mutual information (Hindle, 1990). Historically, such early distributional representations provided *explicit* (that is, directly meaningful and human interpretable) information: the features of such vectors were composed by, e.g., binary values, or probabilistic measures (Navigli & Martelli, 2019). The number of dimensions of such vectors was determined by the size of the vocabulary.

On the other side, in *implicit* or latent representations, features were used resulting from Latent Semantic Analysis (LSA). LSA is a multidimensional associative model based on the distributional hypothesis: word meaning is encoded as a multi-dimensional (usually 300 or 400 dimensions) vector obtained by elaborating large *corpora* to estimate the co-occurrence frequencies for each word. In order to assess the quality of vectors built through LSA, such representations have been assessed on synonymy tests in (Landauer & Dumais, 1997), finding that these embeddings performed comparably to school-aged children, when measuring similarity between word pairs as the cosine similarity between their corresponding embeddings. Given the interesting features of word embeddings, further energies

have been invested in building vectorial representation through neural networks, in particular it has been shown how NLMs implicitly develop word embeddings when training for the word prediction task (Bengio et al., 2003). The research has rapidly progressed demonstrating that word embeddings could be incorporated into neural architectures for various NLP tasks (Collobert & Weston, 2007, 2008). In particular, the use of pre-trained word vectors for initializing the embedding layer of a task-specific network is an instance of multi-task learning, with language modeling as a supporting task (Goldberg, 2017, p. 243).

Despite the impact of word embeddings on the NLP area, the design of such representation implicitly includes a major limitation: it ignores the fact that words can have multiple meanings and conflates all these meanings into a single representation (Camacho-Collados & Pilehvar, 2018). Such limitation, also referred as Meaning Conflation Deficiency, may affect the semantic understanding of an NLP system that uses word embeddings at its core: word embeddings seem to be unable to grasp different meanings of a word, even if those meanings occur in the training corpus (Schütze, 1998; Yaghoobzadeh & Schütze, 2016). Additionally, the meaning conflation may affect the semantic modeling of word senses, for example two semantically unrelated words similar to different word senses of the same word may be pulled together (Neelakantan, Shankar, Passos, & McCallum, 2014a; Pilehvar & Collier, 2016). In order to alleviate the impact of the meaning conflation deficiency several directions have been taken, one of these research directions is building word embeddings as representations sensitive to the context, what are called contextualized word embeddings. In contrast to word embeddings which represents words with a single static vector, contextualized word embeddings dynamically change depending on the context in which they appear. We will report on the milestones of both kinds of representation in Chapter 3.

2.2.2 Semantic Networks

Another popular approach to characterize word senses is the adoption of semantic networks. Semantic networks are knowledge bases in which the core unit, the *synset*, represents a uniquely identified sense, which mostly reflects cognitively grounded uses of a given term. Synsets are usually represented as vertices of a

labeled graph, where edges represent semantic relationships intervening between each two senses. In contrast to the conflated word embedding representations, semantic networks contain unique entries for different word senses, thus constituting the reference word sense dictionary, i.e. *sense inventory*, for many diverse NLP applications.

WordNet WordNet constitutes the most popular English word sense inventory, manually curated by experts (G. A. Miller, 1995). Word senses are represented through synsets, that are sets of synonyms expressing distinct word senses. In WordNet, each lemma (word or multi-word expression), belongs to one or more synsets, and word senses are represented as combination of word form, i.e. the lexicalization, and synset (usually referred to as sense-key). These nodes are provided with a unique identifier, called synset id, and endowed with a gloss and various usage examples. For example, given the word *bank*, we might intend the word as the river bank or the financial institution, depending on the context of usage. The river bank word sense entry is defined as *sloping land (especially the slope beside a body of water)*, the synset is made of the term *bank* only and is identified by the synset identifier wn09213565n. Together with the definition two examples are reported: *they pulled the canoe up on the bank* and *he sat on the bank of the river and watched the currents*. The financial institution word sense is represented by the synset *depository financial institution, banking concern, banking company, bank* identified by the id wn08420278n and defined as *a financial institution that accepts deposits and channels the money into lending activities*. The reported usage examples are *he cashed a check at the bank* and *that bank holds the mortgage on my home*. [2]

WordNet is actually partitioned into four categories, modeled upon the four open-class parts of speech: nouns, verbs, adjectives and adverbs. Each portion of WordNet has its own relations connecting entities herein. Nouns are organized in a lexical memory as hierarchies, verbs are organized by a variety of entailment relations, while adjectives and adverbs are organized as N-dimensional hyperspaces: each of these lexical structures reflects a different way of categorizing experience. The WordNet version 3.0 contains $117,659$ synsets, $206,949$ senses (sensekeys) and $147,306$ different lemmas. Following WordNet several energies have been invested

[2] We refer to WordNet v. 3.0, available at http://wordnet-rdf.princeton.edu/lemma/bank.

in developing semantic networks or translating WordNet in different languages (Bond & Paik, 2012; J. P. McCrae, Rademaker, Rudnicka, & Bond, 2020; J. P. McCrae, Wood, & Hicks, 2017; Pianta, Bentivogli, & Girardi, 2002; Rudnicka, Witkowski, & Kaliński, 2015).

BabelNet BabelNet is a multilingual lexicalized semantic network, containing about about 20 million entries and distributed in 500 different languages (Navigli & Ponzetto, 2010).[3] The architecture of BabelNet is borrowed from WordNet: the network was built by automatically linking Wikipedia pages to WordNet synsets, thus exploiting the multilingual features of Wikipedia: each BabelNet's node contains multilingual lexicalizations for the same word sense, collected from Wikipedia.

More precisely, BabelNet's generation apporach may be partitioned into three steps: synsets mapping, multilingual expansions and synsets linking. In the first step, WordNet and Wikipedia are combined by automatically acquiring a mapping between synsets and Wikipedia pages: the conditional probability of a WordNet synset given a Wikipedia page is computed through disambiguation contexts obtained from the two resources. The precision of the first step is fundamental to avoid duplicate word senses as well as building solid foundations to the multilingual expansion. The second step is aimed at extending English synsets to multiple languages through both Wikipedia and machine translation. In this setting, each synset, identified through a BabelNet synset id, is enriched with lexicalizations in multiple languages, thus representing each word sense as a collection of lemmas in many different languages. The last step is aimed at building the network between synsets. Relationships are inherited from WordNet and further expanded by considering the degree of correlation between the two Wikipedia pages connected to both nodes. The final resource consists in a semantic network in which nodes (BabelNet synsets) offer multilingual lexicalizations and are linked by all the WordNet relationships plus an underspecified relatedness relation inherited by the Wikipedia page links. Further works have been focused on injecting in BabelNet other information extracted from other resources such as WordNet

[3]The figures are referred to the BabelNet version 5.0, available at https://babelnet.org.

2020 (J. P. McCrae et al., 2020), Omegawiki [4], Wiktionary[5], Wikidata (Vrandečić & Krötzsch, 2014), GeoNames [6], ImageNet (Deng et al., 2009), Open Multilingual WordNet (Bond & Paik, 2012), BabelPic (Calabrese, Bevilacqua, & Navigli, 2020) and VerbAtlas (Di Fabio, Conia, & Navigli, 2019).

[4] http://www.omegawiki.org/
[5] https://www.wiktionary.org/
[6] http://www.geonames.org

3 Related Work

In this Chapter we introduce the state-of-the-art approaches to word and sense embeddings. The Chapter is organized as follows: in Section 3.1 the main contributions to word embeddings are illustrated; both static word embeddings (Section 3.1.1) and context sensitive word embeddings (Section 3.1.2) are surveyed. The second part of the chapter (Section 3.2) elaborates on the main contributions to sense embeddings : in Section 3.2.1 we review static sense embeddings, whilst in Section 3.2.2 context sensitive sense embeddings are introduced.

3.1 Word Embeddings

Given the relevance of the embeddings in lexical resources as well as for neural networks, several energies have been invested in finding an efficient approach to embed words into vectorial representations. As mentioned, according to their underlying constructive rationale, word embeddings might be partitioned into *static* word embeddings and *contextual* word embeddings. Since both families of word embeddings are relevant to our work, in this section we briefly report on both design paradigms.

3.1.1 Static Word Embeddings

One of the main contributions in word embedding techniques is provided by Word2Vec, the innovation lies in making the learning of word embedding efficient, enabling training of word embeddings on large-scale corpora (Mikolov, Chen, et al., 2013). Word2Vec has been published in two different fashions: Skip-gram and Continuous Bag-Of-Words (CBOW). The difference among the two strategies lies in the training objective: in the CBOW setting, given the context surrounding a word w_1, w_{i-1} and w_{i+1}, w_n, we have to predict the word at position w_i; conversely, in

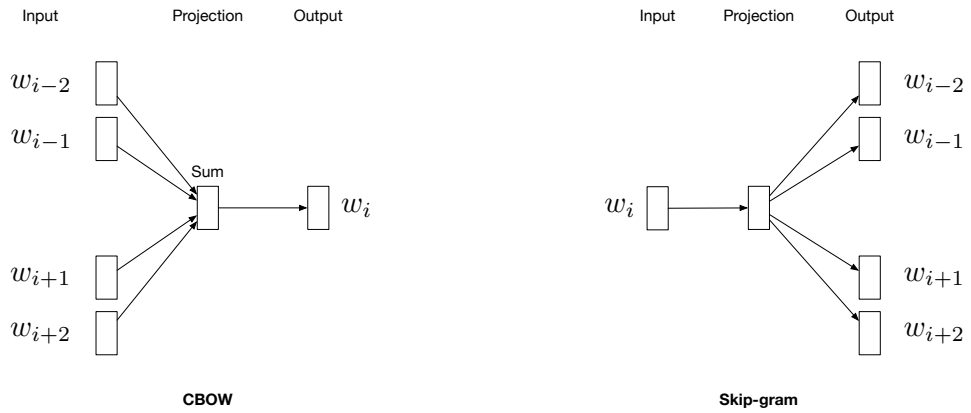

Figure 3.1: Figure from (Mikolov, Chen, et al., 2013). Both CBOW and Skip-gram architectures. CBOW predicts the word w_i given its surrounding context $(w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2})$; Skpi-gram predicts the context $(w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2})$ given the word w_i.

Skip-gram setting, given the word w_i we have to predict the surrounding context w_1, w_{i-1} and w_{i+1}, w_n. Both CBOW and Skip-gram training objectives are graphically illustrated in Figure 3.1. Acquiring word embeddings from large text corpora allows them to incorporate relation among words, such as the relation among country and the relative capital, or the gender of words (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Mikolov, Yih, & Zweig, 2013).

Another word embedding architecture that deserves to be mentioned is GloVe (Pennington, Socher, & Manning, 2014). Such architecture belongs to a different line of research: while Word2Vec model is acknowledged to be a predictive model, GloVe belongs to the count-based family of models: that is, representations are learned by applying dimensionality reduction techniques to the co-occurrence counts matrix. In particular GloVe embeddings have been acquired through a training on 840 billion words from the Common Crawl dataset [1].

One of the latest contributions in prediction-based static word embeddings field is fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017). Such model improves over skip-gram architecture by learning N-gram embeddings rather than word embeddings. The intuition underlying this decision is that language relies heavily on morphology and compositional word-building encodes information also in subwords, so this may be generalized to unseen words (Joulin, Grave, Bojanowski, & Mikolov, 2016).

A different line of research goes in the direction of improving pre-trained word

[1] http://commoncrawl.org.

embeddings by exploiting knowledge bases (Faruqui & Dyer, 2014). Such technique, called retrofitting, improves vectors quality in a post-processing step that updates word representations by running a belief-propagation algorithm on a graph constructed from lexicon-derived relational information. More precisely, given a set of pre-trained word vectors $\hat{Q} = \{\hat{q}_1, \ldots, \hat{q}_n\}$ such technique is aimed at refining such descriptions by accounting for the information provided by a knowledge base. In particular, given a vocabulary $V = \{w_1, \ldots, w_n\}$, an ontology Ω that encodes the semantic relations between words in V and the set of pre-trained word vectors $\hat{Q}$ for words in V, the objective of the retrofitting is to learn the word vector representations $Q = \{q_1, \ldots, q_n\}$ such that the representation q_i for the word w_i is close to its counterpart $\hat{q}_i \in \hat{Q}$ and adjacent to the neighbours of w_i in the ontology Ω. The refined word vectors are obtained through a learning process, since the objective is to achieve a word representation q_i close to its pre-trained vector $\hat{q}_i$ as well as its neighbours q_j $\forall j$ such that the words w_i and w_j are connected by a semantic relation $(i, j) \in \Omega$, the objective to be minimize is:

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i ||q_i - \hat{q}_i||^2 + \sum_{(i,j)} \in \Omega \beta_{ij} ||q_i - q_j||^2 \right]$$

where α and β tune the balance between the pre-trained vector and the ontology association factors. It is worth noting that the selection of the vector distance metric is not bounded, in this setting the Euclidean Distance has been adopted. The retrofitting technique shown improvements in pre-trained word vectors quality, in particular, the authors proved its beneficial effect on word similarity, syntactic relation extraction, synonym selection and sentiment analysis. The retrofitting is also at the core of ConceptNet Numberbatch (CNN) (Speer & Chin, 2016). In particular, CNN is built exploiting the "expanded retrofitting", which adjusts the values of existing word embeddings based on a new objective function that also takes a knowledge graph into account: the authors applied the retrofitting separately to multiple sources of embeddings, i.e. GloVe, Word2Vec and fastText adopting ConceptNet as knowledge base (Speer et al., 2017): the results are then aligned on a unified semantic space. More precisely, the authors exploited ConceptNet as Ω knowledge base, where only non negative relations have been retained —i.e., neg-

ative relations such as *NotUsedFor* or *Antonym* have been removed—. Additionally, the vocabulary V was constructed based on the different vector resources: since multiple pre-trained embedding resources were adopted, the authors took all terms appearing in the first $500,000$ rows of each such resource, and retained all the words occurring in the first $200,000$ rows of at least one of them. Such terms are then combined with the set of words contained in the Ω knowledge base. Subsequently, each vectorial resource was refined though the expanded retrofitting technique so as to obtain the unified embedding matrix M_1. More precisely, for each source of embeddings $\hat{Q}$ (Word2Vec, GloVe) the authors obtained its inferred version Q by applying retrofitting, then such representations were concatenated to the unified embedding matrix M_1. Given that multiple word representations were concatenated in the unified matrix, each word is represented through a vector $q_i \in \mathbb{R}^k$, that is, each embedding is represented with k dimensions corresponding to features that may be redundant. Therefore, the representations from M_1 were subject to dimensionality reduction: such process was designed to learn a projection from k dimensions to $k' = 300$ to remove the redundancy stemming from the concatenation. Additionally, to deal with multiple languages, the authors calculated their own multilingual distributional embeddings through fastText for words occurring in the OpenSubtitles2016 parallel corpus (Lison & Tiedemann, 2016) and employed such vectors as input together with Word2Vec and GloVe. The latest version of CNN [2] covers 78 different languages.

3.1.2 Contextualized Word Embeddings

As mentioned, despite their impact, word embeddings suffer from the meaning conflation deficiency. In order to address such issue two parallel lines of research have emerged: modeling sense representations and building context sensitive word representations. In this section we focus on the latter family.

One of the pioneering works employing contextualized representation is the sequence tagger from W. Li and McCallum (2005). Such model derives context sensitive representations for each word based on word clustering, then integrates them as additional features to a sequence tagger. The attention to context sensitive rep-

[2] https://github.com/commonsense/conceptnet-numberbatch.

resentations remained latent until the lights were turned on the limitations of word embeddings. One of the earliest attempts to address the meaning conflation issue is Context2Vec (Melamud, Goldberger, & Dagan, 2016). Such model represents the context of a target word by extracting the output embedding of a multi-layer perceptron built on top of a bi-directional LSTM language model. Context2Vec started the research line in which contextualized embeddings are pre-trained on large unlabeled data. At the test time word contextualized embeddings are then combined to their static embedding and fed to the main model. Following such direction, the prominent ELMo (Embeddings from Language Models) technique (Peters et al., 2018) built context sensitive word embeddings through the pre-training of a bi-directional language model on large text corpus. The difference with respect to preceding works is that ELMo jointly maximizes the log likelihood of the forward and backward directions, thus sharing and combining weights from both directions. Similarly to ELMo, the Context Vectors (CoVe) model computes contextualized representations using a two-layer bidirectional LSTM network, in the machine translation setting: CoVe vectors are pre-trained exploiting an LSTM encoder from an attentional sequence-to-sequence machine translation model (McCann, Bradbury, Xiong, & Socher, 2017). The introduction of Transformers architecture (Vaswani et al., 2017) started two varieties of NLMs: predictive and bidirectional language models (more on this in Chapter 2). In particular, bidirectional NLMs such as BERT, exploit the encoder of Transformers to build context sensitive word embeddings. Such models are pre-trained on more and more larger text corpus, thus obtaining representation as much general as possible. Pre-trained language models can subsequently be exploited to compute representations for words in sentences: the embeddings produced by such models are then used to address many diverse NLP tasks in many different languages (Lee et al., 2020; Qu et al., 2019; Raffel et al., 2019; Souza, Nogueira, & Lotufo, 2019).

3.2 Sense Embeddings

In order to alleviate the meaning conflation deficiency of word embeddings, a parallel direction of research has emerged over the past years, which tries to directly model individual meanings of words. In this section we focus on sense represen-

tations relying on word embeddings. Since we partitioned word embeddings in *static* and *contextualized* representations, we therefore partitioned sense representations according to the underlying word embeddings design paradigm. Additionally, attempting to address the meaning conflation limitation, mainly two lines of sense vector representations have emerged: in the first line, unsupervised, senses are learned directly from text corpus or knowledge-based (E. H. Huang, Socher, Manning, & Ng, 2012; Reisinger & Mooney, 2010; Vu & Parker, 2016), while in the second approach senses are linked to pre-defined sense inventories. In this work we focus on the latter type of representation.

3.2.1 Static Sense Embeddings

Provided that the evolution of word representations flows from static word embeddings to contextualized word embeddings, the first vector representations for word senses have been introduced relying on static word embeddings.

One of the main contributions between the sense vectorial representations is NASARI (Camacho-Collados, Pilehvar, & Navigli, 2015b; Pilehvar & Navigli, 2015). In the same spirit of BabelNet, NASARI puts together two sorts of knowledge: one coming from WordNet (originally handcrafted by a team of lexicographers), based on synsets and on the intervening semantic relations, and one available in Wikipedia, which is conversely the outcome of a large collaborative effort. Pages in Wikipedia are considered as concepts. In NASARI embeddings each item (concept or named entity) is defined through a dense vector over a 300-dimensions space. NASARI vectors have been acquired by starting from the vectors trained on the Google News dataset, provided along with the Word2vec toolkit. All NASARI vectors share the same semantic space also with Word2vec, so that their representations can be used to compute semantic distances between any two such vectors. Thanks to the structure provided by the BabelNet resource, the resulting $2.9M$ embeddings are part of a huge semantic network. NASARI includes sense descriptions for nouns, but not for other grammatical categories.

Directly following NASARI, but with totally different building rationale, SENSEEMBED has been introduced, containing representations for the four main parts of speech (nouns, verbs, adjectives and adverbs) (Iacobacci, Pilehvar, & Navigli, 2015).

The approach proposed by SENSEEMBED is aimed at obtaining continuous representations of individual senses. In order to build sense representations, the authors exploited Babelfy (Moro, Raganato, & Navigli, 2014) to disambiguate the September-2014 dump of the English Wikipedia.[3] Subsequently, the Word2vec toolkit has been employed to build vectors for 2.5 millions of unique word senses. The obtained resource contains the representation for both terms —e.g., the embedding for the term Bank— and word senses —e.g., the embedding representing the meaning of bank intended as *financial institution*, endowed with the identifier Bank-bn:00008364n—.

Given the negative impact of the meaning conflation deficiency implicitly coded in word embeddings, DECONF has been introduced with particular attention to the mentioned limitation (Pilehvar & Collier, 2016). DECONF is a sense representation technique that starts from a semantic network and a set of pre-trained word embeddings. The proposed approach computes a list of "sense biasing words" for a given word sense. The whole process is characterised by two phases: *(i)* the extraction of the most representative words that express the semantics of a synset, and *(ii)* the sense representations learning. In the extraction phase, the control strategy starts from a target synset y_t, leverages the structure of the semantic network of WordNet and produces as output an ordered list $\mathcal{B}_t$ of semantically related terms that provide a cue for the sense usage. The latter phase is aimed at learning the representation of the word sense s_t (the sense for term t): to these ends the procedure deconflates the representation of all the lexicalizations of the sense s_t, and biases them towards the list $\mathcal{B}_t$. In order to generate the DECONF resource, the authors chose WordNet 3.0 as semantic network and the 300-*d* Word2Vec word embeddings trained on the Google News dataset. The final resource contains about 207 thousand vectors for WordNet word senses, each such sense representation lives in the same space which is also shared by the word embeddings.

Different from previous resources, SW2V (so named after 'Senses and Words to Vectors') is a neural model devised to represent both term and sense vector representations (Mancini, Camacho-Collados, Iacobacci, & Navigli, 2017). The proposed approach jointly learns both representations by exploiting text corpora and seman-

[3] http://dumps.wikimedia.org/enwiki/.

tic networks. Due to the temporal complexity of the state-of-the-art disambiguation systems, the authors devised an unsupervised shallow word sense connectivity algorithm. Such algorithm exploits the connections of a semantic network and associates a term with its top candidate senses according to the number of sense connections and word context. Once the corpus of sense tagged words has been generated, an extension of the Word2Vec's CBOW model is employed. The extension of the CBOW model in order to deal with word senses follows the assumption that since a word is a lexicalization of an underlying sense, an update of the word embedding should entail a similar update of the sense representation, and vice versa. The authors chose BabelNet as reference semantic network and its underlying sense inventory; the pre-trained version of SW2V contains over 6 million vectors representing both words and word senses, in the same spirit of SENSEEMBED.

Following SW2V, LSTMEMBED is a recently proposed model based on bidirectional LSTM for learning embeddings of words and senses in the same semantic space (Iacobacci & Navigli, 2019). The model starts from a sense-tagged text, which is processed with a bidirectional LSTM analyzing both the preceding and the posterior context of a token s_i, where s_i is either a word or a sense tag. The output computed by the LSTM on both directions is concatenated and linearly weighted with a dense layer. Subsequently the model compares the output with the pre-trained embedding vector of the target token s_i. The training phase maximizes the similarity among the output of the network and the pre-trained embeddings: the loss is computed in terms of cosine distance.[4] LSTMEMBED pre-trained embeddings contain about 2 millions vectors. The obtained resource is featured by three sorts of representation: the word-sense representation —e.g., the vector for the sense bn:00008363n, which refers to Bank, intended as "*Sloping land, especially the slope beside a body of water*"—; the representation for a given lexicalization associated to a given sense —e.g., for the pair Bank-bn:00008363n—; and the word embedding —e.g., the vector for the term Bank—, possibly conflating all senses underlying the given term.

[4] In order to generate the LSTMEMBED pre-trainied embeddings the authors chose the BabelNet 4.0 sense inventory. The BabelWiki corpus (Scozzafava, Raganato, Moro, & Navigli, 2015) has been employed for both the training of the model and the representation of the objective embeddings; the latter case has been addressed using the Word2Vec's SkipGram model.

3.2.2 Contextualized Sense Embeddings

The contextualized sense representations line of research follows directly from the introduction of contextualized language models, and strongly relies on such representations. Despite such language models provide context sensitive representations, they still lack semantic grounding to sense inventories.

The first attempt at demonstrating that contextual embeddings from pre-trained language models can be enriched by exploiting sense inventories is Language Modelling Makes Sense (LMMS) (Loureiro & Jorge, 2019a). LMMS is an approach for generating sense embeddings relying on pre-trained contextualized language models that covers the entire WordNet 3.0 sense inventory. The proposed approach computes a list of sense embeddings starting from annotations, i.e. a sense tagged corpus. In particular, the sense vector is computed as the average of all the contextual representation for words tagged with the word sense: given n contextual embeddings c_i for a word sense s, the vector v_s is computed as $v_s = \frac{1}{n}\sum_{i=1}^{n} c_i$. Since the sense tagged corpus covers only a small percentage of the WordNet vocabulary, the authors improve sense inventory coverage exploiting WordNet structure: in order to build embeddings for higher-level abstractions, the average of the embeddings of all lower-level constituents is employed. That is, the embedding of an unseen word sense corresponds to the average of all its children representation. LMMS pre-trained embeddings cover the entire WordNet vocabulary, thus containing embeddings for $117,659$ synsets corresponding to $206,949$ unique senses. Since LMMS is grounded to the WordNet sense inventory, the resource represent vectors for English words only.

Following LMMS, SENSEMBERT has been introduced relying on the pre-trained version of BERT large (Scarlini, Pasini, & Navigli, 2020a). SENSEMBERT is a knowledge-based approach to produce latent semantic representations of word meanings in multiple languages. The construction of SENSEMBERT relies on Babelnet, Wikipdia and NASARI sense embeddings together with the pre-trained BERT large model. The proposed approach starts by collecting from Wikipedia all the sentences that are suitable for characterizing a given word synset: this is done by exploiting the link between BabelNet and Wikipedia. Once contextual information has been collected, the authors compute the contextualized word embedding of

each relevant word for the target synset: relevant words for each synset are identified exploiting NASARI lexical vectors, then contextualized representation for such words are obtained through BERT large language model. Eventually, the synset embedding is built by exploiting word representations together with their rank in the NASARI lexical vector. In the same spirit of LMMS, synset representation quality is improved by exploiting the semantic network structure. Since the linking between BabelNet and Wikipedia involves nouns only, the proposed approach build representations for nouns only. Thanks to the multilingual nature of BabelNet, the authors exploited also the multilingual version of BERT to build sense embeddings for multiple languages. SENSEMBERT pre-trained embeddings contain vectors for $146,313$ senses.

ARES, so dubbed after context-AwaRe Embeddings of Senses, has been introduced few months later, as the extension of SENSEMBERT (Scarlini et al., 2020b). ARES is a semi-supervised approach to producing sense embeddings for the lexical meanings within a lexical knowledge base that lie in a space that is comparable to that of contextualized word vectors. The construction of ARES relies on several resources: WordNet, SyntagNet (Maru, Scozzafava, Martelli, & Navigli, 2019), UKB (E. Agirre, de Lacalle, & Soroa, 2014) and BERT. The proposed approach starts collecting contexts for WordNet's synsets exploiting BERT: given a sense s and one of its lexicalizations l, the authors collected all occurrences of l in a corpus and computed their contextualized representation and clustered through *k-means*. To such groups the UKB algorithm is exploited so as to label each cluster with one of the senses for l. Each such cluster is then refined exploiting the collocations from SyntagNet. Contextual information retrieved is then exploited so as to build embeddings for WordNet's synsets as a combination of embeddings computed though BERT for sentences and collocations. ARES pre-trained embeddings contain vectors for $206,950$ senses, covering 65% of WordNet's vocabulary ($77,195$ out of $117,659$).

LMMS Reloaded (LMMS-R) is the most recent resource belonging to the contextualized sense embeddings family, it has been introduced as extension of LMMS (Loureiro, Jorge, & Camacho-Collados, 2022). LMMS-R is a principled approach for sense representation based on contextual NLMs trained exclusively with self-

supervision. Following LMMS the synset embbedding for s is built by averaging the contextualized representations for the lexicalization l in a sense-tagged corpus. Such vectors are then refined exploiting the WordNet structure. LMMS-R allows for a different characterization of multiple layers NLMs according to the task for which the embeddings are designed. LMMS-R pre-trained embeddings for Word Sense Disambiguation (WSD) cover the entire WordNet vocabulary, thus containing embeddings for $117,659$ synsets corresponding to $206,949$ unique senses. Since LMMS-R is grounded to the WordNet sense inventory, the resource represents vectors for English words only.

4 LessLex

LessLex (Linking multilingual Embeddings to SenSe representations of LEXical items) is the first resource that we developed (Colla, Mensa, & Radicioni, 2020a), consisting of a set of distributional vectors built by merging BabelNet and ConceptNet Numberbatch. In this Chapter we will show that the adoption of distributional sense representations can be beneficial to the resolution of various NLP tasks.

The Chapter is organized as follows: in Section 4.1 we introduce the main control strategy to build LessLex: we start by illustrating the procedure to select the seed terms from the sense inventory (Section 4.1.1), then we present the approaches aimed at extending such initial representations (Section 4.1.2); and finally, we report on figures and features that describe LessLex embeddings (Section 4.1.3). Afterwards, in Section 4.2 we introduce the evaluation of LessLex vectors, reporting on three different tasks: word similarity (Section 4.2.1), contextual word similarity (Section 4.2.2) and semantic text similarity (Section 4.2.3). We conclude the Chapter by discussing the results obtained in the whole experimentation (Section 4.2.4).

4.1 Building LessLex

The generation of LessLex relies on two resources: BabelNet and CNN. We briefly recall them for the sake of self-containedness. BabelNet is a wide-coverage multilingual semantic network resulting from the integration of lexicographic and encyclopedic knowledge from WordNet and Wikipedia. Word senses are represented as *synsets* which are uniquely identified by Babel Synset identifiers (e.g., `bn:03739345n`). Each synset is enriched by further information about that sense, such as its possible lexicalizations in a variety of languages, its gloss (a brief description) and its Wikipedia Page Title. Moreover, it is possible to query BabelNet to retrieve all the meanings (synsets) for a given term. While the construction of BabelNet is by de-

sign essential to our approach, in principle we could plug different sets of word embeddings. We chose CNN word embeddings as our starting point for a number of reasons, and namely: its vectors are to date highly accurate; all such vectors are mapped onto a single shared multilingual semantic space spanning over 78 different languages; it ensures reasonable coverage for general purposes use (Speer & Lowry-Duda, 2017); also, it allows dealing in uniform way with multi-word expressions, compound words (Havasi, Speer, & Alonso, 2007), and even flexed forms; it is released under the permissive MIT License.

The algorithm for the generation of LESSLEX is based on an intuitive idea: to exploit multilingual terminological representations in order to build precise and punctual conceptual representations. Without loss of generality, we introduce our methodology by referring to nominal senses, while the whole procedure also applies to verb and adjectival senses, so that in the following we will switch between sense and concept as appropriated. Each concept in LESSLEX is represented by a vector generated by averaging a set of CNN vectors. Given the concept c, we retrieve it in BabelNet to obtain the sets $\{\mathcal{T}^{l_1}(c), \ldots, \mathcal{T}^{l_n}(c)\}$ where each $\mathcal{T}^{l}(c)$ is the set of lexicalizations in the language l for c.[1] We then try to extract further terms from the concepts' English gloss and English Wikipedia Page Title (WT from now on). The final result is the set $\mathcal{T}^{+}(c)$ that merges all the multilingual terms in each $\mathcal{T}^{l}(c)$ plus the terms extracted from the English gloss and WT. In $\mathcal{T}^{+}(c)$ we retain only those terms that can be actually found in CNN, so that the LESSLEX vector $\vec{c}$ can be finally computed by averaging all the CNN vectors associated to the terms in $\mathcal{T}^{+}(c)$.

4.1.1 Selecting the sense inventory: seed terms

Since the generation algorithm creates a representation for conceptual elements (be them nominal, verbal or adjectival senses), it is required to define which concepts will be hosted in the final resource. For this purpose we define a set of terms that we call *seed terms*. Seed terms are taken from different languages and different POS (nouns, verbs and adjectives are presently considered), and their meanings

[1] We presently consider all the languages that are adopted during the evaluation: English (eng), French (fra), German (deu), Italian (ita), Farsi (fas), Spanish (spa), Portuguese (por), Basque (eus) and Russian (rus).

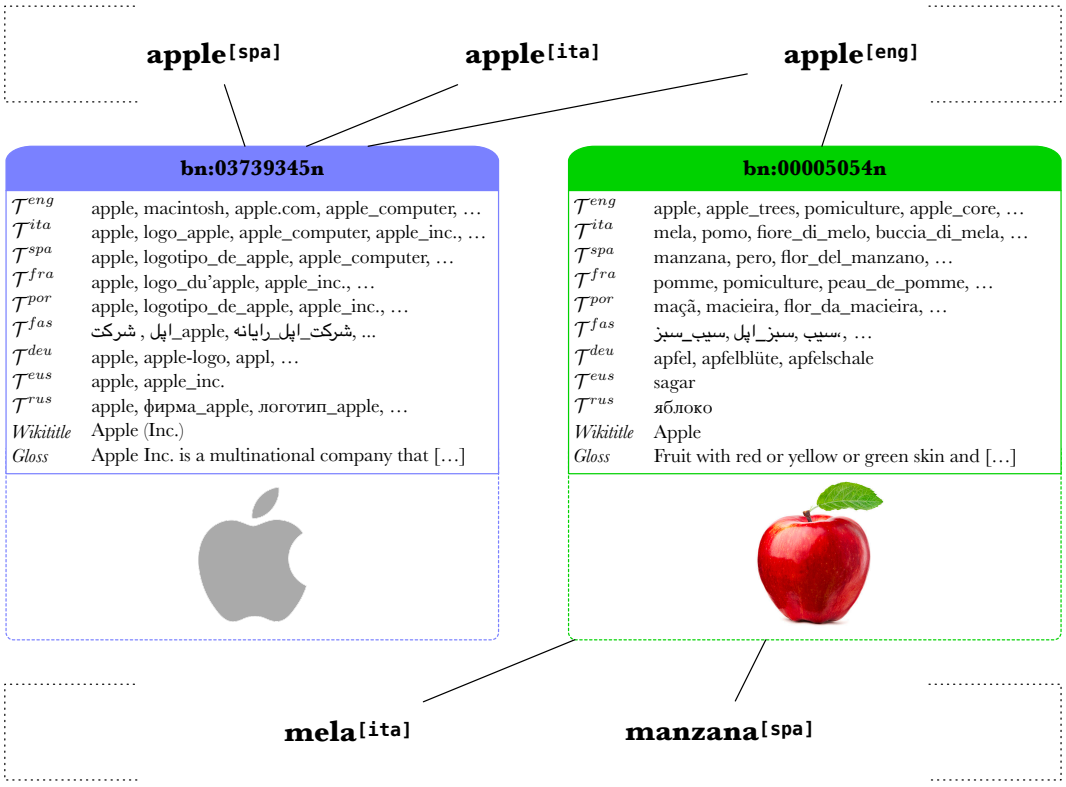

Figure 4.1: Retrieval of two senses for five seed terms in three different languages.

(retrieved via BabelNet) constitute the set of senses described by LESSLEX vectors. Due to the polysemy of language and to the fact that the seed terms are multilingual, different seed terms can retrieve the same meaning. Seed terms do not affect the generation of a vector, but they rather determine the coverage of LESSLEX, since they are used to acquire the set of concepts that will be part of the final resource. Figure 4.1 illustrates this process for a few seed terms in English, Spanish and Italian. These terms provide two senses in total: `bn:03739345n` – *Apple (Inc.)* and `bn:00005054n` – *Apple (fruit)*. The first one is the meaning for $apple^{spa}$, $apple^{ita}$ and $apple^{eng}$, while the second one is a meaning for $manzana^{spa}$, $mela^{ita}$ and, again, $apple^{eng}$. Each synset contains all the lexicalizations in all languages, together with the English gloss and the WT. This information will be exploited for building $\mathcal{T}^{+}(c_{bn:03739345n})$ and $\mathcal{T}^{+}(c_{bn:00005054n})$ during the generation process.

4.1.2 Extending the set of terms

As anticipated, we not only rely on the lexicalizations of a concept to build its $\mathcal{T}^{+}$, but we also try to include further specific words, parsed from its English gloss and

WT. The motivation behind this extension is the fact that we want to prevent $\mathcal{T}^+$ from containing only one element: in such case, the vector for the considered sense would coincide with that of the more general term, possibly conflating different senses. In other words, enriching $\mathcal{T}^+$ with further terms is necessary to reshape vectors that have only one associated term as lexicalization. For instance, starting from the term $sunset^{eng}$ we encounter the sense `bn:08410678n` (representing the city of Sunset, Texas). This sense is provided with the following lexicalizations:

$$\mathcal{T}^{eng} = \{sunset^{eng}\}; \quad \mathcal{T}^{spa} = \{sunset^{spa}\}; \quad \mathcal{T}^{fra} = \{sunset^{fra}\}.$$

However, out of these three terms only $sunset^{eng}$ actually appears in CNN, giving us a final singleton $\mathcal{T}^+ = \{sunset^{eng}\}$. At this point no average can be performed, and the final vector in LESSLEX for this concept would be identical to the vector of $sunset^{eng}$ in CNN. Instead, if we take into consideration the gloss ''*Township in Starr County, Texas*'', we can extract $township^{eng}$ and append it in $\mathcal{T}^+$, thus obtaining a richer vector for this specific sense of *sunset*. In the following Sections we describe the two strategies that we developed in order to extract terms from WTs and glosses. The extension strategies are applied for every concept, but in any case, if the final $\mathcal{T}^+$ contains a single term ($|\mathcal{T}^+| = 1$), then we discard the sense and we do not include its vector in LESSLEX.

Extension via Wikipedia Page Title

The extension via WT only applies to nouns, since senses for different POS are not present in Wikipedia. In detail, if the concept has a Wikipedia Page attached and if the WT provides a disambiguation or specification (e.g., *Chips (company)* or *Magma, Arizona*) we extract the relevant component (by exploiting commas and parentheses of the Wikipedia naming convention) and search for it in CNN. If the whole string cannot be found, we repeat this process by removing the leftmost word of the string until we find a match. In so doing, we search for the maximal sub-string of the WT that has a description in CNN. This allows us to obtain the most specific and yet defined term in CNN. For instance, for the WT *Bat (guided bomb)* we may not have a match in CNN for *guided bomb*, but we can at least add *bomb* to the set of terms in $\mathcal{T}^+$.

Table 4.1: List of the extraction rules in a regex style, describing some POS patterns. If a gloss or a portion of a gloss matches the left part of the rule, then the elements in the right part are extracted. Extracted elements are underlined.

	Nouns		
1.	to be NN+	⟶	NN+
2.	NN1 CC NN2	⟶	NN1,NN2
3.	DT ∗ NN+	⟶	NN+
	Verbs		
1.	to be VB	⟶	VB
2.	Sentence starts with a VB	⟶	VB
3.	VB1 ((CC | ,) VB2)+	⟶	VB1, VB2+
	Adjectives		
1.	Sentence is exactly JJ	⟶	JJ
2.	*not* JJ	⟶	(JJ is dropped)
3.	(*relate*|*relating*|*related*) *to* ∗ NN	⟶	NN
4.	JJ1 CC JJ2	⟶	JJ1,JJ2
5.	JJ1, JJ2 *or* JJ3	⟶	JJ1, JJ2, JJ3

Extension via gloss

Glosses often contain precious pieces of information that can be helpful in the augmentation of the terms associated to a concept. We parse the gloss and extract its components. By construction, descriptions provided in BabelNet glosses can originate from either WordNet or Wikipedia (Navigli & Ponzetto, 2012). In the first case we have (often elliptical) sentences, such as (`bn:00028247n` – *door*) "a swinging or sliding barrier that will close the entrance to a room or building or vehicle". On the other side, Wikipedia typically provides a plain description like "A door is a panel that makes an opening in a building, room or vehicle". Thanks to the regularity of these languages, with few regular expressions on POS patterns[2] we are able to collect enough information to enrich $\mathcal{T}^+$. We devised several rules according to each sense POS; the complete list is reported in Table 4.1. As an example, from the following glosses we extract the terms in bold (the matching rule is shown in square brackets):

[2]We adopted the Penn Treebank POS set: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

- [Noun-2] bn:00012741n (*Branch*) *A **stream** or **river** connected to a larger one.*

- [Noun-3] bn:00079944n (*Winner*) *The **contestant** who wins the contest.*

- [Noun-1] bn:01276497n (*Plane (river)*) *The Plane is a **river** in Brandenburg, Germany, left tributary of the Havel.*

- [Verb-2] bn:00094850v (*Tee*) ***Connect** with a tee.*

- [Verb-3] bn:00084198v (*Build*) ***Make** by **combining** materials and parts.*

- [Adjective-3] bn:00106822a (*Modern*) *Relating to a recently developed **fashion** or style.*

- [Adjective-4] bn:00103672a (*Good*) *Having **desirable** or **positive** qualities especially those suitable for a thing specified.*

In Figure 4.2 we provide an example of the generation process for three concepts, provided by the seed terms $gate^{eng}$ and $gate^{ita}$. For the sake of simplicity, we only show the details regarding two languages (English and Italian). Step *(1)* shows the input terms. In step *(2)* we retrieve three meanings for $gate^{eng}$ and one for $gate^{ita}$, which has already been fetched since it is also a meaning for $gate^{eng}$. For each concept we collect the set of lexicalizations in all considered languages, plus the extensions extracted from WT and gloss. We then merge all such terms in $\mathcal{T}^+$, by retaining only those that can be actually found in CNN. Once the $\mathcal{T}^+$ sets are computed, we access CNN to retrieve the required vectors for each set *(3)* and then we average them, finally obtaining the vectors for the concepts at hand *(4)*.

4.1.3 LESSLEX features

We now describe the main features of LESSLEX, together with the algorithm to compute conceptual similarity on this resource. The final space in which LESSLEX vectors reside is an extension of the CNN multilingual semantic space. Each original CNN vector co-exists with the set of vectors that represent its underlying meanings. This peculiar feature allows us to compute the distance between a term and each of its corresponding senses, and such distance is helpful to determine, given a pair of terms, in which sense they are intended. For example, in assessing the

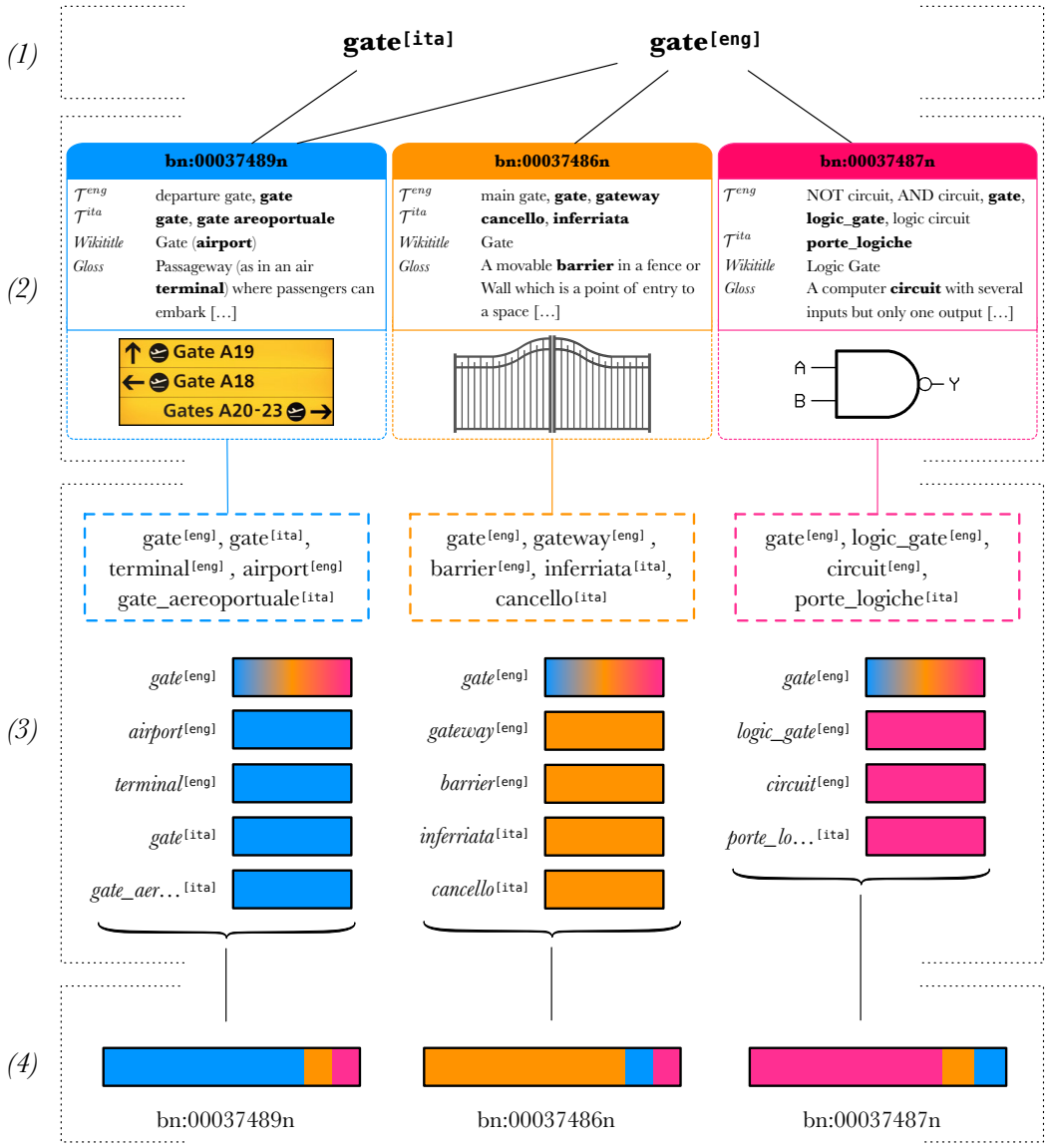

Figure 4.2: Generation of three LESSLEX vectors, starting from the seed terms $gate^{eng}$ and $gate^{ita}$.

similarity of two terms such as 'glass' and 'eye', most probably the recalled senses would differ from those recalled for the pairs 'glass' and 'window', and 'glass', 'wine'.

LESSLEX Statistics

The LESSLEX resource[3] has been generated from a group of seed terms collected by starting from 56, 322 words taken from the Corpus of Contemporary American En-

[3] LESSLEX can be downloaded at the URL https://ls.di.unito.it/resources/lesslex/.

Table 4.2: Figures on the generation process of LESSLEX, divided by Part of Speech

LESSLEX Statistics	All	Nouns	Verbs	Adjectives
Seed terms	84,620	45,297	11,943	27,380
Terms in BabelNet	65,629	41,817	8,457	15,355
$\mathcal{T}^+$ avg. cardinality	6.40	6.16	9.67	6.37
Discarded Senses	16,666	14,737	368	1,561
Unique Senses	174,300	148,380	11,038	14,882
Avg. senses per term	4.80	6.12	3.77	1.77
Total extracted terms	227,850	206,603	8,671	12,576
Avg. extracted terms per call	1.40	1.46	1.06	1.05

glish (COCA) (Davies, 2009),[4] 19,789 terms fetched from the relevant dictionaries of the Internet Dictionary Project[5] and the 12,544 terms that appear in the datasets that we used during the evaluation. All terms were POS tagged and duplicates removed beforehand. The final figures of the resource and details concerning its generation are reported in Table 4.2.

We started from a total of 84,620 terms, and for 65,629 of them we were able to retrieve at least one sense in BabelNet. The $\mathcal{T}^+$ cardinality shows that our vectors were built by averaging about 6 CNN vectors for each concept. Interestingly, verbs seem to have much richer lexical sets. The final number of senses in LESSLEX amounts to 174,300, with a vast majority of nouns. We can also see an interesting overlap between the group of senses associated to each term. If we take nouns as example, we have around $42K$ terms providing $148K$ unique senses (3.5 per term), while the average polysemy per term counting repetitions amounts to 6.12. So, we can observe that approximately three senses per term are shared with some other term. A huge amount of concepts are discarded since they only have one term inside $\mathcal{T}^+$: these are named entities or concepts with poor lexicalization sets. The extraction process provided a gran total of about $228K$ terms, and on average each $\mathcal{T}^+$ contains 1.40 additional terms extracted from Wikipedia Page Titles and glosses.

Out of the $117K$ senses in WordNet (version 3.0), roughly $61K$ of them are cov-

[4]COCA is a *corpus* covering different genres, such as spoken, fiction, magazines, newspaper and academic (http://corpus.byu.edu/full-text/).
[5]http://www.june29.com/idp/IDPfiles.html.

ered in LESSLEX. It is however important to note that additional LESSLEX vectors can be built upon any set of concepts, provided that they are represented in BabelNet (which contains around $15M$ senses) and that some of their lexicalizations are covered in CNN ($1.5M$ terms for the considered languages).

Computing word similarity: maximization and ranked-similarity

The word similarity task consists in computing a numerical score that expresses how similar two given terms are. Vectorial resources such as CNN can be easily employed to solve this task: in fact, since terms are represented as vectors, the distance (usually computed through cosine similarity, or some other variant of angular distance) between the two vectors associated to the input terms can be leveraged to obtain a similarity score. While terminological resources can be directly employed to compute a similarity score between words, conceptually grounded resources (e.g., NASARI, LESSLEX) do not allow directly computing word similarity, but rather *conceptual similarity*. In fact, such resources are required to determine which senses must be selected while computing the score for the terms. In most cases this issue is solved by computing the similarity between all the combinations of senses for the two input terms, and then by selecting the maximum similarity as the result score (Pedersen, Banerjee, & Patwardhan, 2005). In formulae, given a term pair $\langle t_1, t_2 \rangle$ and their corresponding list of senses $s(t_1)$ and $s(t_2)$, the similarity can be computed as

$$\text{sim}(t_1, t_2) = \max_{\vec{c}_i \in s(t_1), \vec{c}_j \in s(t_2)} [\text{sim}(\vec{c}_i, \vec{c}_j)] \tag{4.1}$$

where $\text{sim}(\vec{c}_i, \vec{c}_j)$ is the computation of conceptual similarity employing the vector representation for the concepts at hand.

To compute the conceptual similarity between LESSLEX vectors we have devised a different approach, which we call *ranked similarity*. Since we are able to determine not only the distance between each two senses of the input terms, but also the distance between each input term and all of its senses, we use this information to fine tune the computed similarity scores and use ranking as a criterion to grade senses relevance. In particular, we hypothesise that the relevance of senses for a given term can be helpful for the computation of similarity scores, so we de-

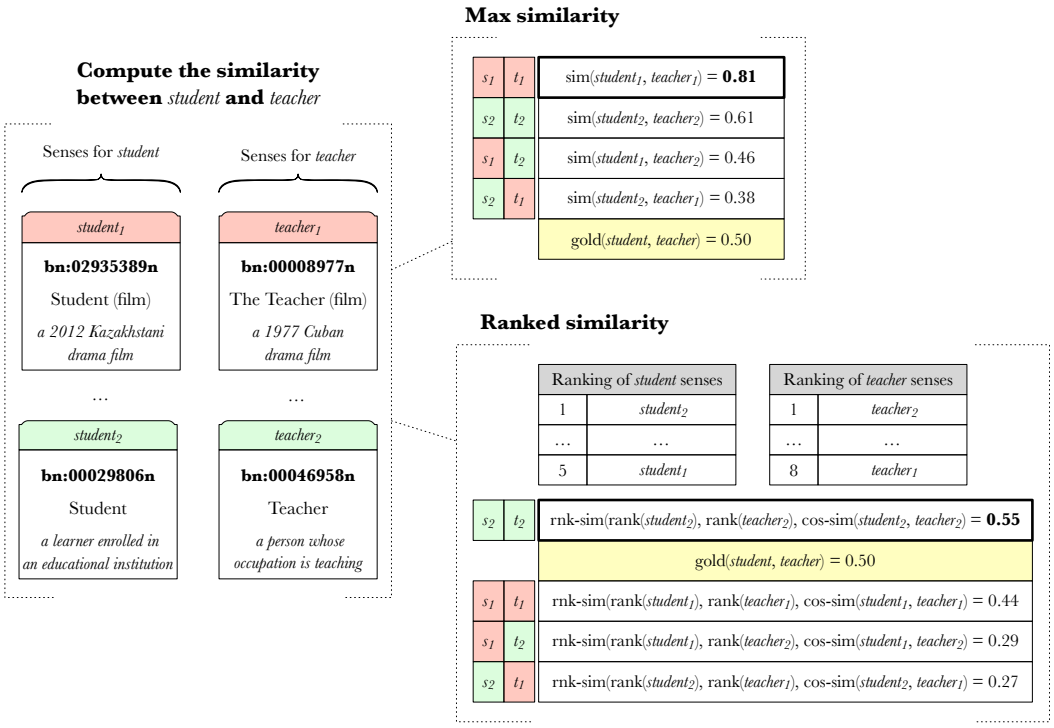

Figure 4.3: A comparison between the max-similarity (Equation 4.1) and the ranked-similarity (Equation 4.2) approaches for the computation of the conceptual similarity.

vised a measure that also accounts for the *ranking* of distances between senses and seed term. It implements a heuristics aimed at considering two main elements: the relevance of senses (senses closer to the seed term are preferred), and similarity between sense pairs. Namely, the similarity between two terms t_1, t_2 can be computed as:

$$\text{rnk-sim}(t_1, t_2) = \max_{\substack{\vec{c}_i \in s(t_1) \\ \vec{c}_j \in s(t_2)}} \left[\left((1-\alpha) \cdot (\text{rank}(\vec{c}_i) + \text{rank}(\vec{c}_j))^{-1} \right) + \left(\alpha \cdot \text{cos-sim}(\vec{c}_i, \vec{c}_j) \right) \right], \quad (4.2)$$

where α is used to tune the balance between ranking factor and raw cosine similarity.[6] We illustrate the advantages of the ranked similarity with the following example (Figure 4.3). Let us consider the two terms *teacher* and *student*, whose gold-standard similarity score is 0.50.[7] One of the senses of teacher is `bn:02193088n` (*The Teacher (1977 film)* - a 1977 Cuban drama film) while one of the senses of stu-

[6] Presently $\alpha = 0.5$.

[7] We borrow this word pair from the SemEval 17 Task 2 dataset (Camacho-Collados, Pilehvar, Collier, & Navigli, 2017).

dent is bn:02935389n (*Student (film)* - a 2012 Kazakhstani drama film). These two senses have a cosine similarity in LESSLEX of 0.81: such a high score is reasonable, since they are both drama movies. However, it is clear that an annotator would not refer to these two senses for the input terms, but rather to bn:00046958n (*teacher* - a person whose occupation is teaching) and bn:00029806n (*student* - a learner who is enrolled in an educational institution). These two senses obtain a similarity score of 0.61, which will not be selected since it is lower than 0.81 (as computed through the formula in Equation 4.1). However, if we take into consideration the similarities between the terms *teacher* and *student* and their associated senses, we see that the senses that one would select —while requested to provide a similarity score for the pair— are much closer to the seed terms. The proposed measure involves re-ranking the senses based on their proximity to the term representation, thereby emphasising more relevant terms. We finally obtain similarity of 0.44 for the movie-related senses, while the school-related senses pair obtains a similarity of 0.55, which will be selected and better correlates with human rating.

Since the ranked-similarity can be applied only if both terms are available in CNN (so that we can compute the ranks among their senses), we propose a twofold setup for the usage of LESSLEX. In the first setup we only make use of the ranked-similarity, so in this setting if at least one given term is not present in CNN we discard the pair as not covered by the resource. In the second setup (LESSLEX-OOV, designed to deal with *Out Of Vocabulary* terms) we implemented a fallback strategy to ensure higher coverage: in this case, in order to cope with missing vectors in CNN, we adopt the max-similarity as similarity measure in place of the ranked-similarity.

4.2 Evaluating LESSLEX

In order to assess the flexibility and quality of our embeddings we carried out a set of experiments involving both intrinsic and extrinsic evaluation. Namely, we considered three different tasks:

1 the Semantic Similarity task, where two terms or —less frequently— senses are compared and systems are asked to provide a numerical score express-

ing how close they are; systems' output is compared to human ratings (Section 4.2.1);

2 the more recent Contextual Word Similarity task, asking systems to either assess the semantic similarity of terms taken in context (rather than as pairs of terms taken in isolation), or to decide whether a term has same meaning in different contexts of usage (Section 4.2.2); and

3 the Semantic Text Similarity task, where pairs of text excerpts are compared to assess their overall similarity, or to judge whether they convey equal meaning or not (Section 4.2.3).

4.2.1 Word Similarity Task

In the first experiment we tested LESSLEX vectors on the word similarity task: linguistic items are processed in order to compute their similarity, which is then compared against human similarity judgement. Word similarity is mostly thought of as closeness over some metric space, and usually computed through cosine similarity, although different approaches exist, e.g., based on cognitively plausible models (Jimenez, Becerra, Gelbukh, Bátiz, & Mendizábal, 2013; Lieto, Mensa, & Radicioni, 2016a; Mensa, Radicioni, & Lieto, 2017; Tversky, 1977). We chose to evaluate our word embeddings on this task because it is a relevant one, for which many applications can be drawn such as Machine Translation (Lavie & Denkowski, 2009), Text Summarization (Mohammad & Hirst, 2012) and Information Retrieval (Hliaoutakis, Varelas, Voutsakis, Petrakis, & Milios, 2006). Although this is a popular and relevant task, until recently it has been substantially limited to monolingual data, often in English. Conversely, we collected and experimented on all major cross-lingual datasets.

Experimental setting

In this Section we briefly introduce and discuss the selection of datasets adopted for the evaluation.

A pioneering dataset is WordSim-353 (Finkelstein et al., 2002); it has been built by starting from two older sets of word pairs, the RG-65 and MC-30 datasets (G. A. Miller

Table 4.3: List of the dataset employed in the experimentation, showing the POS involved and the languages available in both monolingual and cross-lingual versions.

Dataset	Part of Speech	Monolingual	Cross-lingual
RG-65[1]	nouns	eng, fas, spa	eng, spa, fas, por, fra, deu
WS-Sim-353[2]	nouns	eng, ita, deu, rus	-
SimLex-999[3]	nouns, verbs, adjectives	eng, ita, deu, rus	-
SimVerbs-3500[4]	verbs	eng	-
SemEval 17[5]	nouns	eng, deu, ita, spa, fas	eng, deu, ita, spa, fas
Goikoetxea [6]	nouns, verbs, adjectives	eus	eng, eus, spa, ita

[1] http://lcl.uniroma1.it/similarity-datasets/, https://www.seas.upenn.edu/~hansens/conceptSim/.
[2] http://www.leviants.com/ira.leviant/MultilingualVSMdata.html.
[3] https://fh295.github.io/simlex.html, http://www.leviants.com/ira.leviant/MultilingualVSMdata.html.
[4] http://people.ds.cam.ac.uk/dsg40/simverb.html.
[5] http://alt.qcri.org/semeval2017/task2/index.php?id=data-and-tools.
[6] http://ixa2.si.ehu.es/ukb/bilingual_embeddings.html.

& Charles, 1991; Rubenstein & Goodenough, 1965). These dataset were originally conceived for the English language and compiled by human experts. They have then been translated to multilingual and to cross-lingual datasets: the RG-65 has been translated into Farsi and Spanish by Camacho-Collados, Pilehvar, and Navigli (2015a), while the WordSim-353 has been translated by Leviant and Reichart (2015b) into Italian, German and Russian through crowdworkers fluent in such languages. Additionally, WordSim-353 has been partitioned by individuating the subset of word pairs appropriate for experimenting on similarity judgements rather than on relatedness judgements (E. Agirre et al., 2009). The SimLex-999 dataset has been compiled through crowdsourcing, and includes English word pairs covering different parts of speech, namely nouns (666 pairs), verbs (222 pairs) and adjectives (111 pairs) (Hill, Reichart, & Korhonen, 2015). It has been then translated into German, Italian and Russian by Leviant and Reichart (2015a). A dataset has been proposed entirely concerned with English verbs, the SimVerbs-3500 dataset (Gerz, Vulić, Hill, Reichart, & Korhonen, 2016); similar to SimLex-999, items herein have been obtained from the USF free-association database (Nelson, McEvoy, & Schreiber, 2004). The SemEval-17 dataset has been developed by Camacho-Collados et al. (2017); it contains many uncommon entities, like *Si-o-seh pol* or *Mathematical Bridge* encompassing both multilingual and cross-lingual data. Finally, another dataset has been recently released by Goikoetxea, Soroa, and Agirre (2018), in the following referred to as Goikoetxea dataset, built by adding further cross-lingual versions

for the RG-65, WS-WordSim-353 and SimLex-999 datasets.

In our evaluation both multilingual and cross-lingual translations have been used. A *multilingual* dataset is one (like RG) where term pairs $\langle x, y \rangle$ from language i have been translated as $\langle x', y' \rangle$ into a different language, such that both x' and y' belong to the same language. An example is $\langle casa, chiesa \rangle$, $\langle house, church \rangle$, or $\langle maison, église \rangle$. Conversely, in a cross-lingual setting (like SemEval 2017, Task 2 - cross-lingual subtask), x' is a term from a language different from that of y', like in the pair $\langle casa, church \rangle$.

Many issues can afflict any dataset, as it is largely acknowledged in literature (Camacho-Collados et al., 2017, 2015a; Hill et al., 2015; E. H. Huang et al., 2012). The oldest datasets are too small (in the order of few tens of word pairs) to attain full statistic significance; until recent years, typically similarity and relatedness (association) judgements have been conflated, thereby penalising models concerned with similarity. Additionally, for such datasets the correlation between systems' results and human rating is higher than human inter-rater agreement. Since human ratings are largely acknowledged as the upper bound to artificial performance in this kind of task, it has been raised that such datasets are not fully reliable benchmarks to investigate the correlation between human judgement and systems' output. Furthermore, a tradeoff exists between the size of the dataset and the quality of the annotation: resources acquired through human experts annotation typically are more limited in size, but featured by higher inter-rater agreement (in the order of .80), while larger datasets suffer from a lower (often with $< .7$) agreement among annotators, thus implying overall reduced reliability. We thus decided to test on all main datasets adopted in literature, to provide the most comprehensive evaluation, widening the experimental base as much as possible. The most recent datasets are in principle more controlled and reliable —SimLex-999, SimVerbs, SemEval-2017, Goikoetxea—, but still we decided to experiment on all of them, since even RG-65 and WS-Sim 353 have been widely used until recently. All benchmarks employed in the experiments are illustrated in Table 4.3.

We have then selected a set of recent and influential sense and word embeddings from the literature, and used them for experimentation. In the following we provide a brief description for each such resource. ConceptNet Numberbatch is

Table 4.4: List of the resources considered in the experimentation and the algorithm we employed for the resolution of the word similarity task.

	Description	Algorithm
LL-M	LESSLEX	mf-sense similarity
LL-O	LESSLEX (strategy for handling OOV terms)	ranked-similarity
LLX	LESSLEX	ranked-similarity
CNN [1]	ConceptNet Numberbatch word embeddings	cosine similarity
NAS [2]	NASARI sense embeddings	max similarity
JCH [3]	JOINTCHYB bilingual word embeddings	cosine similarity
SSE [4]	SENSEEMBED sense embeddings	max similarity
N2V [5]	NASARI sense embeddings + Word2Vec word embeddings	ranked-similarity

[1] Speer et al. (2017) (http://github.com/commonsense/conceptnet-numberbatch v. 16.09)
[2] Camacho-Collados, Pilehvar, and Navigli (2016) (http://lcl.uniroma1.it/nasari/ v. 3.0)
[3] Goikoetxea et al. (2018) (http://ixa2.si.ehu.es/ukb/bilingual_embeddings.html)
[4] Iacobacci et al. (2015) (http://lcl.uniroma1.it/sensembed/)
[5] Word2Vec embeddings trained on UMBC (http://lcl.uniroma1.it/nasari/)

a set of pre-trained word embeddings built by integrating vector representations from Word2Vec, GloVe and fastText with the commonsense knowledge from ConceptNet (Speer & Chin, 2016). More precisely, the retrofitting technique has been applied to each such word vector resource so as to obtain knwoledge-oriented word representations. Subsequently, the word vectors have been concatenated and compressed through dimensionality reduction, thus obtaining a single 300-dimensional representation for each word in the vocabulary. Additionally, by exploiting multilingual distributional embeddings from fastText the authors built vector representations for words in many different languages. The latest version of ConceptNet Numberbatch covers 78 different languages.

JOINTCHYB stems form a novel bilingual word embedding method based on the representation of words in a joint space, as well as the use of bilingual constraints and bilingual synthetic corpora, both derived from bilingual wordnets, in the learning process (Goikoetxea et al., 2018). More precisely, the proposed approach is based on a random walk algorithm over bilingual wordnets which produces lexicalizations in two languages as it traverses the wordnets. The obtained bilingual corpus is combined with monolingual corpora and is then fed into the Skip-Gram model, yielding bilingual embeddings. Further improvements were obtained by incorporating bilingual constraints extracted from the wordnets into the Skip-Gram loss function. In order to learn textual embeddings for the target

languages the authors used Wikipedia corpora[8] except for Basque: in fact, the embeddings for Basque language are based on the Wikipedia dump (2016/04/07) combined to the Elhuyar Web Corpus (Leturia, 2012). Regarding wordnets, the authors employed the Multilingual Central Repository (A. G. Agirre, Laparra, Rigau, & Donostia, 2012) and the ItalWordNet (Roventini et al., 1998) for Italian language.

In the same spirit of BabelNet, NASARI puts together two sorts of knowledge: one coming from WordNet (originally handcrafted by a team of lexicographers), based on synsets and on the intervening semantic relations, and one available in Wikipedia, which is conversely the outcome of a large collaborative effort (Camacho-Collados et al., 2016). Pages in Wikipedia are considered as concepts. In NASARI embeddings each item (concept or named entity) is defined through a dense vector over a 300-dimensions space. NASARI vectors have been acquired by starting from the vectors trained on the Google News dataset, provided along with the Word2vec toolkit. All NASARI2VEC vectors share the same semantic space also with Word2vec, so that their representations can be used to compute semantic distances between any two such vectors. Thanks to the structure provided by the BabelNet resource, the resulting $2.9M$ embeddings are part of a huge semantic network.

The approach proposed by SENSEEMBED is aimed at obtaining continuous representations of individual senses (Iacobacci et al., 2015). In order to build sense representations, the authors exploited Babelfy (Moro et al., 2014) to disambiguate the September-2014 dump of the English Wikipedia.[9] Subsequently, the Word2vec toolkit has been employed to build vectors for 2.5 millions of unique word senses. The obtained resource contains the representation for both terms —e.g., the embedding for the term Bank— and word senses —e.g., the embedding representing the meaning of bank intended as *financial institution*, endowed with the identifier Bank-bn:00008364n—.

The results obtained by employing LESSLEX and LESSLEX-OOV are compared to those obtained by employing NASARI and CNN, to elaborate on similarities and differences with such resources. Additionally, we report the correlation indices obtained by experimenting with other word and sense embeddings that either are

[8] http://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/
[9] http://dumps.wikimedia.org/enwiki/.

trained to perform on specific datasets (JOINTCHYB by Goikoetxea et al. (2018)), or that directly compare to our resource, as containing both term-level and sense-level vector descriptions (SENSEEMBED and NASARI2VEC). Table 4.4 summarizes the considered resources and the algorithm used to compute the semantic similarity. In these respects, we adopted the following rationale. When testing with resources that allow for a combined use of word and sense embeddings we use the ranked-similarity[10] (as described in Equation 4.2); when testing with sense embeddings we adopt the max similarity/closest senses strategy (Budanitsky & Hirst, 2006; Pilehvar & Navigli, 2015; Resnik, 1995) to select senses; while handling word embeddings we make use of the cosine similarity, by borrowing the same approach as illustrated in (Camacho-Collados et al., 2017).[11] In order to provide some insights on the quality of the ranked-similarity, we also experiment on an algorithmic baseline referred to as LL-M (LESSLEX Most Frequent Sense), where we selected the most frequent sense of the input terms based on the connectivity of the considered sense in BabelNet. The underlying rationale is, in this case, to study how this strategy to pick up senses compares with LESSLEX vectors, that are built from word embeddings that usually tend to encode the most frequent sense of each word. Finally, in the case of RG-65 dataset concerned with sense labeled pairs (Schwartz & Gomez, 2011)[12] we only experimented on sense embeddings, and the similarity scores have been computed through the cosine similarity metrics.

Results

All tables report Pearson and Spearman correlations (denoted by r and ρ, respectively); dashes indicate that a given resource does not deal with the considered

[10] In the experimentation α was set to 0.5.

[11] A clarification must be done about SENSEEMBED. Since in this resource both terminological and sense vectors co-exist in the same space, the application of the ranked-similarity would be fitting. However, in SENSEEMBED every sense representation is actually indexed on a pair $\langle term, sense \rangle$, so that different vectors may correspond to a given *sense*. In the ranked-similarity, when computing the distance between a term t and its senses, we retrieve the sense identifiers from BabelNet, so to obtain from SENSEEMBED the corresponding vector representations. Unfortunately, however, most senses s_i returned by BabelNet have no corresponding vector in SENSEEMBED associated to the term t (i.e., indexed as $\langle t, s_i \rangle$). This fact directly implies a reduced coverage, undermining the performances of SENSEEMBED. We then realized that the ranked-similarity is an unfair and not convenient strategy to test on SENSEEMBED (in that it forces to use it to some extent improperly), so we resorted to using the max similarity instead.

[12] This version of the RG-65 dataset has been sense-annotated by two humans with WordNet 3.0 senses.

Table 4.5: Results on the multilingual and cross-lingual RG-65 dataset, consisting of 65 word pairs. As regards as monolingual correlation scores for the English language, we report results for similarity computed by starting from terms (at *words* level), as well as results with sense identifiers (marked as *senses*). The rest of the results were obtained by using word pairs as input. Reported figures express Pearson (r) and Spearman (ρ) correlations.

RG-65	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
[Word] eng	.64	.59	**.91**	.86	**.91**	.86	**.91**	**.90**	.67	.67	.84	.86	.75	.81	.80	.75
[Sense] eng	-	-	**.94**	**.91**	**.94**	**.91**	-	-	.81	.76	-	-	.72	.76	.78	.73
fas (N)	.75	.72	.75	.75	.73	.70	**.76**	**.76**	.58	.50	-	-	.66	.66	-	-
spa (N)	.82	.82	**.93**	**.93**	**.93**	**.93**	.92	**.93**	.88	.87	.80	.84	.82	.85	-	-
por-fas (N)	.71	.69	.85	.85	.81	.79	**.87**	**.86**	.52	.62	-	-	.70	.66	-	-
fra-por (N)	.82	.83	.92	**.89**	.92	**.89**	**.93**	.88	.69	.67	-	-	.81	.74	-	-
fra-fas (N)	.73	.72	.84	.84	**.86**	.84	**.86**	**.85**	.47	.58	-	-	.72	.71	-	-
fra-spa (N)	.81	.80	**.93**	**.91**	**.93**	**.91**	**.93**	.89	.79	.82	-	-	.88	.86	-	-
fra-deu (N)	.81	.84	**.90**	**.89**	**.90**	**.89**	.88	.87	.77	.77	-	-	.77	.75	-	-
spa-por (N)	.83	.83	**.93**	**.91**	**.93**	**.91**	**.93**	**.91**	.75	.79	-	-	.79	.79	-	-
spa-fas (N)	.71	.70	**.86**	**.87**	.82	.80	**.86**	.86	.50	.64	-	-	.72	.79	-	-
eng-por (N)	.74	.71	**.94**	**.90**	**.94**	**.90**	.92	**.90**	.78	.77	-	-	.80	.76	-	-
eng-fas (N)	.67	.62	**.86**	.85	.84	.81	**.86**	**.87**	.47	.56	-	-	.73	.71	-	-
eng-fra (N)	.71	.70	**.94**	**.92**	**.94**	**.92**	.92	.91	.76	.73	-	-	.81	.75	-	-
eng-spa (N)	.72	.71	**.93**	**.93**	**.93**	**.93**	**.93**	.92	.85	.85	.83	.86	.80	.85	-	-
eng-deu (N)	.74	.72	**.91**	**.89**	**.91**	**.89**	.89	**.89**	.70	.74	-	-	.76	.80	-	-
deu-por (N)	.87	.84	**.91**	**.87**	**.91**	**.87**	**.91**	**.87**	.73	.76	-	-	.76	.72	-	-
deu-fas (N)	.77	.74	.85	**.85**	**.87**	.84	.85	.84	.58	.65	-	-	.78	.80	-	-
deu-spa (N)	.84	.85	**.91**	**.90**	**.91**	**.90**	.90	.89	.71	.79	-	-	.79	.80	-	-

input, either because lacking of sense representation, or because lacking of cross-lingual vectors. Similarity values for uncovered pairs were set to the middle point of the similarity scale. Additionally, in Appendix A.1 we report the results obtained by considering only the word pairs covered by all the resources: such figures are of interest, since they allow examining the results obtained from each resource 'in purity', by focusing only on their representational precision. All top scores are marked with bold fonts.

Multilingual/Cross-lingual RG-65 dataset The results obtained over the multilingual and cross-lingual RG-65 dataset are illustrated in Table 4.5. RG-65 includes a multilingual dataset and a cross-lingual one. As regards as the former one, both LESSLEX and LESSLEX-OOV obtain analogous correlation with respect to CNN when considering term pairs; LESSLEX and LESSLEX-OOV substantially outperform NASARI, SENSEEMBED and NASARI2VEC while considering sense

Table 4.6: Results on the WS-Sim-353 dataset, where we experimented on the 201 word pairs (out of the overall 353 elements) that are acknowledged as appropriated for computing similarity. Reported figures express Pearson (r) and Spearman (ρ) correlations.

WS-Sim-353	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N)	.67	.65	**.78**	.78	**.78**	.78	**.78**	**.79**	.60	.61	.72	.72	.69	.73	.71	.70
ita (N)	.67	.68	.70	.73	**.74**	**.78**	.69	.73	.66	.65	.60	.62	.66	.73	-	-
deu (N)	.73	.71	.63	.68	.76	.77	**.82**	**.81**	.64	.63	-	-	.62	.60	-	-
rus (N)	.72	.70	.64	.62	**.73**	**.75**	.65	.63	.63	.61	-	-	.60	.60	-	-

pairs (Schwartz & Gomez, 2011). Of course CNN is not evaluated in this setting, since it only includes representations for terms. As regards as the latter subset, containing cross-lingual files, figures show that both CNN and LESSLEX obtained high correlations, higher than the competing resources providing meaning representations for the considered language pairs.

Multilingual WS-Sim-353 dataset The results on the multilingual WS-Sim-353 dataset are presented in Table 4.6. Results on this data differ according to the considered language: interestingly enough, for the English language, the results computed via LESSLEX are substantially on par with those obtained by employing CNN vectors. As regards as the remaining translations of the dataset, CNN and LESSLEX achieve the highest correlations also on the Italian, German and Russian languages. Different from other experimental settings (see, e.g., the RG-65 dataset), the differences in correlation are more consistent, with LESSLEX obtaining top correlation scores for Italian and Russian, and CNN for German.

Multilingual SimLex-999 dataset The results obtained on the SimLex-999 dataset are reported in Table 4.7. We face here twofold results: as regards as the English and the Italian translation, we recorded better results when using the LESSLEX vectors, with consistent advantage over competitors on English verbs. As regards as English adjectives, the highest correlation was recorded when employing the LESSLEX Most Frequent Sense vectors (LL-M column). As regards as Italian, as in the WordSim-353 dataset, the LESSLEX-OOV strategy obtains correlations with human ratings that are higher or on par with respect to those obtained by using LESSLEX vectors. In the second half of the dataset CNN performed better on Ger-

Table 4.7: Results on the multilingual SimLex-999, including overall 999 word pairs, with 666 nouns, 222 verbs and 111 adjectives for the English, Italian, German and Russian languages. Reported figures express Pearson (r) and Spearman (ρ) correlations.

SimLex-999	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N)	.51	.50	**.69**	**.67**	**.69**	**.67**	.66	.63	.40	.38	.55	.53	.52	.49	.46	.43
eng (V)	.62	.56	**.67**	**.65**	**.67**	**.65**	.61	.58	-	-	.51	.50	.54	.49	-	-
eng (A)	**.84**	**.83**	.82	.79	.82	.79	.80	.78	-	-	.63	.62	.55	.51	-	-
eng (*)	.57	.55	**.70**	**.69**	**.70**	**.69**	.67	.65	-	-	.55	.54	.53	.49	-	-
ita (N)	.50	.49	**.66**	**.63**	.64	**.63**	.64	.61	.45	.46	.47	.47	.56	.49	-	-
ita (V)	.58	.52	**.69**	**.63**	**.69**	**.63**	.67	.58	-	-	.54	.47	.54	.44	-	-
ita (A)	.65	.58	**.74**	**.69**	**.74**	**.69**	**.74**	.66	-	-	.39	.30	.57	.47	-	-
ita (*)	.51	.47	**.66**	**.62**	.65	**.62**	.65	.61	-	-	.46	.44	.54	.47	-	-
deu (N)	.58	.56	.65	.63	.65	.64	**.66**	**.65**	.41	.42	-	-	.47	.43	-	-
deu (V)	.48	.42	.54	.45	.54	.46	**.63**	**.57**	-	-	-	-	.43	.37	-	-
deu (A)	.66	.63	.66	.65	.69	.68	**.77**	**.75**	-	-	-	-	.43	.26	-	-
deu (*)	.55	.52	.62	.59	.63	.61	**.67**	**.65**	-	-	-	-	.45	.38	-	-
rus (N)	.43	.42	.52	.48	.51	**.50**	**.53**	.48	.20	.22	-	-	.26	.21	-	-
rus (V)	.31	.19	.25	.18	.27	.20	**.60**	**.55**	-	-	-	-	.23	.20	-	-
rus (A)	.25	.26	.25	.25	.27	.28	**.69**	**.69**	-	-	-	-	.04	.04	-	-
rus (*)	.36	.32	.43	.37	.42	.39	**.56**	**.51**	-	-	-	-	.23	.13	-	-

Table 4.8: Results on the SimVerbs-3500 dataset, containing 3,500 verb pairs. Reported figures express Pearson (r) and Spearman (ρ) correlations.

SimVerbs	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (V)	.58	.56	**.67**	**.66**	**.67**	**.66**	.62	.60	-	-	.56	.56	.45	.42	.31	.30

man and Russian.

SimVerbs-3500 dataset Results obtained while testing on the SimVerbs-3500 dataset are reported in Table 4.8. In this case it is straightforward to notice that the results obtained by LESSLEX outperform those by all competitors, with a gain of .05 in Pearson r, and .06 in Spearman correlation over CNN, on this large set of 3500 verb pairs. It was not possible to use NASARI vectors, that only exist for noun senses; also notably, the results obtained by employing the baseline (LL-M) strategy outperformed those obtained through SENSEEMBED and NASARI2VEC.

Sem Eval 17 Task 2 dataset The figures obtained by experimenting on the "SemEval 17 Task 2: Multilingual and Cross-lingual Semantic Word Similarity" dataset

Table 4.9: Results on the SemEval 17 Task 2 dataset, containing 500 noun pairs. Reported figures express Pearson (r) and Spearman (ρ) correlations.

SemEval 17	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N)	.71	.72	**.79**	.80	.77	**.81**	**.79**	.79	.64	.65	.50	.45	.69	.73	.64	.64
deu (N)	**.73**	.72	.69	.68	.71	**.75**	.70	.68	.62	.62	-	-	.60	.61	-	-
ita (N)	.74	.75	.66	.65	**.76**	**.79**	.63	.61	.72	.73	.54	.50	.70	.73	-	-
spa (N)	**.77**	.79	.67	.66	.74	**.80**	.63	.62	.72	.73	.50	.48	.68	.71	-	-
fas (N)	.67	.67	.43	.47	**.72**	**.75**	.39	.35	.54	.53	-	-	.60	.63	-	-
deu-spa (N)	**.76**	.77	.69	.68	.74	**.79**	.66	.64	.54	.55	-	-	.65	.68	-	-
deu-ita (N)	**.75**	.76	.68	.67	**.75**	**.79**	.65	.63	.53	.65	-	-	.62	.62	-	-
eng-deu (N)	**.75**	.75	**.75**	.75	**.75**	**.79**	.74	.73	.51	.62	-	-	.63	.63	-	-
eng-spa (N)	.75	.76	.73	.73	**.76**	**.82**	.70	.70	.66	.70	.46	.44	.59	.61	-	-
eng-ita (N)	.74	.76	.72	.72	**.76**	**.82**	.69	.69	.63	.71	.38	.36	.69	.73	-	-
spa-ita (N)	**.76**	.77	.67	.66	**.76**	**.81**	.63	.61	.65	.72	.41	.39	.59	.61	-	-
deu-fas (N)	.72	.73	.55	.52	**.73**	**.76**	.51	.47	.39	.52	-	-	.63	.65	-	-
spa-fas (N)	.72	.73	.55	.52	**.75**	**.79**	.50	.47	.47	.61	-	-	.66	.70	-	-
fas-ita (N)	.72	.73	.53	.50	**.75**	**.78**	.49	.45	.43	.58	-	-	.66	.69	-	-
eng-fas (N)	.71	.72	.58	.55	**.74**	**.79**	.54	.51	.42	.59	-	-	.67	.70	-	-

are provided in Table 4.9. This benchmark is a multilingual dataset including 500 word pairs (nouns only) for monolingual versions, and 888 to 978 word pairs for the cross-lingual ones.

These results are overall favourable to LESSLEX in the comparison with CNN and with all other competing resources. Interestingly enough, while running the experiments with CNN vectors we observed even higher correlation scores than those obtained in the SemEval 2017 evaluation campaign (Camacho-Collados et al., 2017; Speer et al., 2017). At that time, such figures scored highest on all multilingual tasks (with the exception of the Farsi language) and on all cross-lingual settings (with no exception). To date, as regards as the cross-lingual setting, LESSLEX correlations indices are constantly higher than those by competitors, including CNN. We observe that the scores obtained by employing the baseline with most frequent senses (LL-M) are always ameliorative with respects to all results obtained by experimenting with NASARI, JOINTCHYB, SENSEEMBED and NASARI2VEC (with the only exception of the ρ score obtained by SSE on the English monolingual dataset).

Table 4.10: Results on the Goikoetxea dataset. The dataset includes variants of the RG-65 (first block), WS-Sim-353 (second block) and SimLex-999 (third block) datasets. The 'eus' abbreviation indicates the Basque language. Reported figures express Pearson (r) and Spearman (ρ) correlations.

Goikoetxea	LL-M		LLX		LL-O		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
spa-eus (N)	.74	.72	.42	.67	**.76**	**.77**	.66	.61	.71	.74	.73	.72	.61	.71	-	-
eng-eus (N)	.74	.74	.41	.77	**.89**	**.91**	.77	.73	**.89**	.88	.88	.87	.81	.83	-	-
eng-spa (N)	.72	.71	**.93**	**.93**	**.93**	**.93**	**.93**	**.93**	.77	.82	.83	.86	.64	.85	-	-
eus-ita (N)	.27	.68	.42	**.74**	.24	.71	.51	.53	.49	.56	**.52**	.58	.20	.58	-	-
spa-ita (N)	.29	.66	.29	**.76**	.29	.74	**.63**	.70	.53	.57	.54	.60	.21	.59	-	-
spa-eus (N)	.31	.74	.40	**.78**	.29	**.78**	.55	.56	.59	.66	**.69**	.73	.23	.64	-	-
eng-ita (N)	.30	.64	.27	**.77**	.32	.76	**.67**	.74	.47	.52	.59	.64	.21	.59	-	-
eng-eus (N)	.30	.70	.39	**.79**	.29	.78	.56	.57	.52	.60	**.71**	.75	.23	.64	-	-
eng-spa (N)	.34	.66	.27	**.79**	.40	.77	**.70**	.76	.52	.56	.68	.73	.29	.64	-	-
eng-spa (N)	.49	.48	**.66**	**.64**	.65	**.64**	.64	.62	.36	.46	.54	.51	.53	.50	-	-
eng-spa (V)	.54	.50	.61	.59	**.62**	**.60**	.58	.56	-	-	.43	.43	.52	.49	-	-
eng-spa (A)	.72	.73	.73	.74	.72	**.75**	**.74**	.74	-	-	.56	.55	.53	.47	-	-
eng-spa (*)	.53	.51	**.66**	.64	.65	**.65**	.64	.63	-	-	.50	.52	.53	.49	-	-
eng-ita (N)	.52	.52	**.70**	**.68**	**.70**	**.68**	.68	.66	.36	.45	.51	.50	.54	.51	-	-
eng-ita (V)	.49	.40	.57	.51	.57	.51	**.67**	**.62**	-	-	.47	.51	.44	.33	-	-
eng-ita (A)	.75	.74	**.79**	**.78**	**.79**	**.78**	.77	.72	-	-	.42	.43	.57	.45	-	-
eng-ita (*)	.50	.46	.65	.62	.65	.63	**.68**	**.66**	-	-	.48	.50	.51	.43	-	-
spa-ita (N)	.53	.53	**.67**	.65	**.67**	**.66**	.66	.64	.34	.45	.45	.45	.54	.52	-	-
spa-ita (V)	.44	.39	.51	.46	.51	.46	**.63**	**.60**	-	-	.42	.44	.43	.34	-	-
spa-ita (A)	.68	.66	**.73**	.71	.72	**.73**	**.73**	.69	-	-	.41	.45	.57	.48	-	-
spa-ita (*)	.49	.46	.61	.58	.61	.59	**.66**	**.64**	-	-	.44	.45	.50	.45	-	-

Multilingual/Crosslingual Goikoetxea dataset The results obtained by testing on the Goikoetxea dataset are reported in Table 4.10. The dataset includes new variants for three popular dataset: three cross-lingual versions for the RG-65 dataset (including the Basque language, marked as 'eus' in the Table); the six cross-lingual combinations of the Basque, Italian and Spanish translations of the WS-Sim-353 dataset; and three cross-lingual translations of the SimLex-999 dataset, including its English, Italian and Spanish translations.

Results are thus threefold. As regards as the first block on the RG-65 dataset, LESSLEX results outperform all competitors (to a smaller extent on versions involving the Basque language), including JOINTCHYB, the best model by Goikoetxea et al. (2018). In the comparison with CNN, LESSLEX vectors achieve better results, with higher correlation for cases involving Basque, on par on the English-Spanish dataset. As regards as the second block (composed of cross-lingual translations

of the WS-Sim-353 dataset), we record that the LESSLEX-OOV strategy obtained the top Spearman correlation scores, coupled to poor Pearson correlation scores; while CNN and JCH obtain the best results as regards as the latter coefficients. As regards as the last block of results in Table 4.10 (containing translations for the SimLex-999 dataset), we first observe that comparing the obtained figures is not simple: we report the figures obtained by Goikoetxea et al. (2018) with no distinction in POS. However, if we focus on results on nouns (two thirds of the SimLex-999 dataset), LESSLEX vectors obtain the best results, while it is not easy to determine whether LESSLEX or CNN vectors provided the overall best results on the other parts of speech.

Discussion

We overall experimented on nine different languages (deu, eng, eus, fas, fra, ita, por, rus, spa) and various cross-lingual combinations. Collectively, such tests constitute a widely varied experimental setting, to the best of our knowledge the largest on the semantic similarity task. The obtained results authorise to state that LESSLEX is at least on par with competing state-of-the-art resources, although we also noticed that some room still exists for further improvements, such as the coverage on individual languages (e.g., Russian and German).

Let us start by considering the results on the multilingual WS-Sim-353 and on the SimLex datasets (Tables 4.6 and 4.7, respectively). The results obtained through LESSLEX always improve on those obtained by employing the sense embeddings by SENSEEMBED and NASARI2VEC, that provide term and sense descriptions embedded in the same semantic space, and are thus closer to our resource. Also the comparison with NASARI is favourable to LESSLEX. In the comparison with CNN, we note that while in the English language LESSLEX and LESSLEX-OOV scores either outperform or closely approach those obtained through CNN, in other languages our vectors suffer from the reduced and less rich sense inventory of BabelNet, that in turn determines a lower quality for our vectors. This can be easily figured if one considers that a less rich synset contains less terms to be plugged into our vectors, thereby determining an overall poorer semantic coverage. The poor results obtained by employing LESSLEX on the German and Russian subsets of the

WS-Sim-353 and SimLex-999 datasets probably stem from this sort of limitation.

A consistent difference between LESSLEX ranked-similarity and the LESSLEX-OOV strategy can be observed when a sense is available in BabelNet, but not the corresponding vector in CNN: the LESSLEX-OOV strategy basically consists in resorting to the maximization approach when —due to the lack of a terminological description associated to the sense at hand— it is not possible to compute the ranked-similarity. This strategy was executed in around 9% of cases ($\sigma = 12\%$) over all datasets, ranging from 0% on verbs in the SimVerbs-3500 dataset, up to around 50% for the Farsi nouns in the SemEval-2017 monolingual dataset. Although not employed often, this strategy contributed in many cases to obtain top scoring results, improving on those computed with plain ranked-similarity with LESSLEX, and also in some cases on CNN and NASARI, as illustrated in both the monolingual and cross-lingual portions of the SemEval-2017 dataset (Table 4.9).

Cases where results obtained through LESSLEX improve over those obtained with CNN are important to assess LESSLEX, in that they confirm that the control strategy for building our vectors is effective, and that our vectors contain precise and high quality semantic descriptions. In this sense, obtaining higher or comparable results by using sense embeddings with respect to using word embeddings (with sense embeddings featuring an increased problem space with respect to the latter ones) is *per se* an achievement. Additionally, our vectors are grounded on BabelNet synset identifiers, which allows addressing each sense as part of a large semantic network, providing further information on senses with respect to the meaning descriptions conveyed through the 300-dimensional vectors. While the LESSLEX-OOV is a run-time strategy concerned with the usage of LESSLEX to compare sense pairs, the quality of our vectors is determined by the enrichment step. More specifically, the coverage of our vectors depends on the strategy devised to build $\mathcal{T}^+$ because the coverage is determined both by the number of term-level vectors, and by the number of sense vectors associated to each term, so that in a sense the coverage of LessLex is determined by the size of $\mathcal{T}^+$. Additionally, we register that the elements added to the extended set $\mathcal{T}^+$ are often of high quality, as proven, for example, by the sense-oriented task of the RG-65 dataset, where senses were assessed (Table 4.5, line 2): in this setting, the correlation indices for LESSLEX

and LESSLEX-OOV vectors score highest over all semantic resources, including NASARI, SENSEEMBED and NASARI2VEC.

Also results achieved while testing on the Goikoetxea dataset seem to confirm that our LL-O strategy allows dealing with languages with reduced (with respect to English) coverage and/or sense inventory in either BabelNet or ConceptNet: in 12 out of the overall 18 tests on this dataset, the LESSLEX-OOV strategy earned at least one top scoring correlation index (either r or ρ, as shown in Table 4.10). The comparison with the recent JOINTCHYB embeddings shows that the adoption of a shared conceptual —multilingual— level can be beneficial and advantageous with respect to building specialised pairs of embeddings.

Less relevant under a cross-lingual perspective, but perhaps relevant in order to fully assess the strengths of our resource, LESSLEX vectors achieved by far highest correlation scores on English verbs (please refer to Table 4.7, line 2 and Table 4.8). The comparison with previous literature seems to corroborate this fact (Gerz et al., 2016): in fact, to the best of our knowledge previous state-of-the-art systems achieved around .624 Spearman correlation (Faruqui & Dyer, 2015; Mrkšić et al., 2016).

In order to further deepen the analysis of results, it is instructive to compare the results reported in Tables 4.5-4.10 with those obtained on the fraction of dataset covered by all considered resources, and provided in Appendix A (Tables A.1-A.6). That is, for each dataset we re-run the experiments for all considered resources by restricting to compare only term pairs actually covered by all resources. We will call this evaluation metrics *CbA condition* hereafter (from 'Covered by All'); as opposed to the case in which a mid-scale similarity value was assigned to uncovered terms, referred to as *MSV condition* in the following (from 'Mid Scale Value'). As mentioned, the CbA condition allows evaluating the representational precision of the resources at stake independent of their coverage, whilst a mixture of both aspects is grasped in the the MSV condition. In the leftmost column of Tables in Appendix A we report the coverage for each test. As we can see, coverage is diverse across datasets, ranging from .61 (averaged on all variants, with a minimum on the Farsi language, in the order of .34 and all translations involving the Farsi) in the SemEval-2017 dataset (Table A.5) to 1.0 in the SimVerbs-3500 dataset (Table A.3).

Table 4.11: The top half Table shows a synthesis of the results obtained in the Mid-Scale similarity Value (MSV) experimental condition, whose details have been illustrated in Tables 4.5-4.10; at the bottom we provide a synthesis of the results obtained in the Covered by All (CbA) experimental condition, illustrated in detail in Tables A.1-A.6.

Mid-Scale similarity Value (MSV) Experimental Condition

	LL-M	LLX	LL-O	CNN	NAS	JCH	SSE	N2V
Spearman ρ	7	32	41	33	1	3	0	0
Pearson r	1	32	50	24	0	0	0	0
Total	8	64	91	57	1	3	0	0

Covered by All (CbA) Experimental Condition

	LL-M	LLX	LL-O	CNN	NAS	JCH	SSE	N2V
Spearman ρ	1	61	-	30	0	0	0	0
Pearson r	2	63	-	22	0	0	0	0
Total	3	124	-	52	0	0	0	0

Other notable cases in which relevant variations in coverage were observed are Russian verbs and adjectives in the SimLex-999 dataset, with .20 and .06 coverage, respectively (Table A.4). In general, as expected, the recorded correlations are improved with respect to results registered for the corresponding (same dataset and resource) test in the MSV setup, although spot pejorative cases were observed, as well (see, e.g., CNN results for Italian adjectives, in the SimLex-999 dataset, reported in Table A.4). For example, if we consider the poorly covered SemEval-2017 dataset, we observe the following rough improvements (average over all translations, and both r and ρ metrics) in the correlation indices: .20 for LESSLEX, .22 for CNN, .09 for NASARI, .30 for JOINTCHYB (that does not cover all translations, anyway), .07 for SENSEEMBED, and .09 for NASARI2VEC (only dealing with nouns).

In order to synthetically examine how the CbA experimental condition affected results with respect to the MSV condition, we adopt a rough index, simply counting the number of test results (we consider as a separate test result each Pearson and each Spearman score in Tables A.1-A.6) where each resource obtained highest scores.[13] We thus count overall 152 tests (15 in the SemEval-2017 dataset, 4 in the WS-Sim-353, 1 in the SimVerbs-3500, 16 in the SimLex-999, 19 in the RG-65, and

[13]Of course we are aware that this is only a rough index, that e.g., does not account for the datasets size (varying from 65 to 3,500 word pairs) or the involved POS, and mixing Pearson and Spearman correlation scores.

21 in the Goikoetxea; for each one we consider as separated r and ρ scores). Provided that in several cases we recorded more than one single resource attaining top scores, the impact of the reduced coverage (CbA condition) vs. MSV condition is presented in Table 4.11. In the MSV condition we have LESSLEX-OOV achieving 91 top scoring results, followed by LESSLEX with 64 and CNN with 57. In the CbA experimental condition, the LESSLEX-OOV strategy was never executed (since only the actual coverage of all resources was considered, and no strategy for handling out-of-vocabulary terms was thus necessary), and LESSLEX obtained 124 top scoring results, against 52 for CNN. In the latter condition there were less cases with a tie. All in all, we interpret the different correlation scores obtained in the two experimental conditions as an evidence that LESSLEX embeddings are featured by good coverage (as suggested by the results obtained in the MSV condition) and lexical precision (as suggested by the results obtained in the CbA condition), improving on those provided by all other resources at stake.

Our approach showed to scale well to all considered languages, under the mild assumption that these are covered by BabelNet, and available in the adopted vectorial resource; when such conditions are met, LESSLEX vectors can be in principle built on a streamlined, on-demand, basis, for any language and any POS.

4.2.2 Contextual Word Similarity Task

As the second test bed we experimented on the contextual word similarity task, which is a variant of the word similarity. In this scenario the target words are taken *in context*, meaning that the input word is given as input together with the piece of text in which they occur. In this setting, systems are required to account for meaning variations in the considered context, so that typical static word embeddings such as Word2Vec, ConceptNet Numberbatch, *etc.* are not able to grasp their mutable, dynamic semantics. We tested on both Stanford's Contextual Word Similarities Datastet (SCWS) (E. H. Huang et al., 2012), and on the more recent Word-in-Context Dataset (WiC) (Pilehvar & Camacho-Collados, 2019).

The SCWS dataset defines the problem as a similarity task, where each input record contains two sentences in which two distinct target words t_1 and t_2 are used. The task requires to provide the pair $\langle t_1, t_2 \rangle$ with a similarity score by taking into

Table 4.12: Some descriptive statistics of the WiC dataset. In particular, the distribution of nouns and verbs, number of instances and unique words across training, development and test-set of the WiC dataset are reported.

Split	Instances	Nouns	Verbs	Unique Words
Training	5,428	49%	51%	1,256
Dev	638	62%	38%	599
Test	1,400	59%	41%	1,184

account the context where the given terms occur. The dataset consists of $2,003$ instances, divided into $1,328$ instances whose targets are a noun pair, 399 a verb pair, 97 adjectival pair, 140 contain a verb-noun pair, 30 contain a noun-adjective pair, and 9 a verb-adjective pair. On the other hand, in the WiC dataset the contextual word similarity problem is cast to a binary classification task: each instance is composed of two sentences in which a specific target word t is used. The employed algorithm has to make a decision on whether t assumes the same meaning or not in the two given sentences. The distribution of nouns and verbs across training, development and test-set is reported in Table 4.12, together with figures on number of instances and unique words.

In the following we report the results obtained on the two datasets by experimenting with LESSLEX and the ranked-similarity metrics. Our results are compared to those reported in literature, and to those obtained by experimenting with NASARI2VEC, which is the only competing resource suitable to implement the ranked similarity along with its contextual variant.

Testing on the SCWS dataset

To test on the SCWS dataset we employed both the ranked-similarity (rnk-sim) and the *contextual* ranked-similarity (c-rnk-sim), a variant devised to account for contextual information. As regards as the latter one, given two sentences $\langle S_1, S_2 \rangle$, we first computed the context vectors $\langle \overrightarrow{ctx}_1, \overrightarrow{ctx}_2 \rangle$ with a bag-of-words approach, that is by averaging all the terminological vectors of the lexical items contained therein:

$$\overrightarrow{ctx}_i = \frac{\sum_{t \in S_i} \vec{t}}{N} \tag{4.3}$$

where N is the number of words in the sentence S_i.

Table 4.13: Results obtained by experimenting on the SCWS dataset. Figures report the *Spearman* correlations with the gold standard divided by part of speech. In the top of table we report our own experimental results, while in the bottom results from literature are provided.

System	ALL	N-N	N-V	N-A	V-V	V-A	A-A
LESSLEX (rnk-sim)	0.695	0.692	0.696	0.820	0.641	0.736	0.638
LESSLEX (c-rnk-sim)	0.667	0.665	0.684	0.744	0.643	0.725	0.524
NASARI2VEC (rnk-sim)	-	0.384	-	-	-	-	-
NASARI2VEC (c-rnk-sim)	-	0.471	-	-	-	-	-
SENSEEMBED[1]	0.624	-	-	-	-	-	-
Huang et al. 50d[2]	0.657	-	-	-	-	-	-
Arora at al.[3]	0.652	-	-	-	-	-	-
MSSG.300D.6K[4]	0.679	-	-	-	-	-	-
MSSG.300D.30K[4]	0.678	-	-	-	-	-	-

[1] Iacobacci et al. (2015)
[2] E. H. Huang et al. (2012)
[3] Arora, Li, Liang, Ma, and Risteski (2018)
[4] Neelakantan, Shankar, Passos, and McCallum (2014b), figures reported from Mu, Bhat, and Viswanath (2017)

The two context vectors are then used to perform the sense rankings for the target words, in the same fashion as in the original ranked-similarity:

$$\text{c-rnk-sim}(t_1, t_2, \overrightarrow{ctx}_1, \overrightarrow{ctx}_2) = \max_{\substack{\vec{c}_i \in s(t_1) \\ \vec{c}_j \in s(t_2)}} \left[\left((1-\alpha) \cdot \left(\underbrace{\text{rank}(\vec{c}_i)}_{\text{w.r.t. } \overrightarrow{ctx}_1} + \underbrace{\text{rank}(\vec{c}_j)}_{\text{w.r.t. } \overrightarrow{ctx}_2} \right)^{-1} \right) + \left(\alpha \cdot \text{cos-sim}(\vec{c}_i, \vec{c}_j) \right) \right]. \tag{4.4}$$

Results The results obtained by experimenting on the SCWS dataset are reported in Table 4.13.[14] In spite of the simplicity of the system employing LESSLEX embeddings, our results overcome those reported in literature, where by far more complex architectures were used.

However, such scores are higher than the agreement among human raters, which can be thought of as an upper bound to systems' performance. The Spearman correlation among human ratings (computed on leave-one-out basis, that is by averaging the correlations between each rater and the average of all other ones) is reportedly of 0.52 for the SCWS dataset (Chi & Chen, 2018; Chi, Shih, & Chen, 2018), which can be considered as a poor inter-rater agreement. Also to some extent sur-

[14] Parameters setting: in rnk-sim and in the c-rnk-sim α was set to 0.5 for both LESSLEX and NASARI2VEC.

Table 4.14: Correlation scores obtained with LESSLEX on different subsets of data obtained by varying standard deviation in human ratings. The reported figures show higher correlation when testing on the most reliable (with smaller standard deviation) portions of the dataset. To interpret the standard deviation values, we recall that the original ratings collected in the SCWS dataset were expressed in the range $[0.0, 10.0]$.

σ	c-rank-sim (r)	rank-sim (r)	nof-items
≤ 0.5	0.83	0.82	39
≤ 1.0	0.85	0.86	82
≤ 1.5	0.85	0.85	165
≤ 2.0	0.82	0.84	285
≤ 2.5	0.68	0.83	518
≤ 3.0	0.68	0.79	903
≤ 3.5	0.67	0.75	1,429
≤ 4.0	0.64	0.71	1,822
< 5.0	0.63	0.69	2,003

prising is the fact that the simple ranked-similarity (rnk-sim), which was intended as a plain baseline, surpassed the contextual ranked-similarity (c-rnk-sim), more suited for this task.

To further elaborate on our results we then re-run the experiment by investigating how the obtained correlations are affected by different degrees of consistency in the annotation. We partitioned the dataset items based on the standard deviation recorded in human ratings, obtaining 9 bins, and re-run our system on these, utilizing both metrics, with same parameter settings as in the previous run. In this case the Pearson correlation indices were recorded, in order to investigate the linear relationship between our output and human ratings. As expected, we obtained higher correlations on the most reliable portions of the dataset, those with smallest standard deviation (Table 4.14).

However, we still found surprising the obtained results, since the rnk-sim metrics seems to be more robust than its contextual counterpart. This is in contrast with literature, where the top scoring metrics, originally defined by Reisinger and Mooney (2010), also leverage contextual information (T. Chen, Xu, He, & Wang, 2015; X. Chen, Liu, & Sun, 2014; E. H. Huang et al., 2012). In particular, the *AvgSim* metrics (which is computed as a function of the average similarity of all prototype pairs, without taking into account the context) is reportedly outperformed by the *AvgSimC* metrics, in which terms are weighted by the likelihood of the word con-

texts appearing in the respective clusters). The *AvgSim* and the *AvgSimC* directly compare to our rnk-sim and c-rnk-sim metrics, respectively. In our results, for the lowest levels of standard deviation (that is, for $\sigma \leq 2$), the two metrics perform in similar way; for growing values of σ we observe a substantial drop of the c-rank-sim, while the correlation of the rnk-sim decreases more smoothly. In these cases (for $\sigma \geq 2.5$) contextual information seems to be less relevant than pair-wise similarity of term pairs taken in isolation.

Testing on the WiC dataset

Different from the SCWS dataset, in experimenting on WiC we are required to decide whether a given term conveys same or different meaning in their context, as in a binary classification task. Context-insensitive word embedding models are expected here to approach a random baseline, while the upper bound, provided by human-level performance, is 80% accuracy.

We run two experiments, one where the contextual ranked-similarity was employed, the other with the Rank-Biased Overlap (Webber, Moffat, & Zobel, 2010). In the former case, we used the *contextual* ranked-similarity (Equation 4.4) as the metrics to compute the similarity score, and we added a similarity threshold to provide a binary answer. In the latter case, we designed another simple schema to assess the semantic similarity between term senses and context. At first we built a context vector (Equation 4.3) to acquire a compact vectorial description of both texts at hand, obtaining two context vectors $\overrightarrow{ctx}_1$ and $\overrightarrow{ctx}_2$. We then ranked all senses of the term of interest (based on the cosine similarity metrics) with respect to both context vectors, obtaining s_1^t and s_2^t, as the similarity ranking of t senses from $\overrightarrow{ctx_1}$ and $\overrightarrow{ctx_2}$, respectively. The Rank-Biased Overlap (RBO) metrics was then used to compare the similarity between such rankings. Given two rankings s_1^t and s_2^t, RBO is defined as follows:

$$\text{RBO}(s_1^t, s_2^t) = (1-p)\sum_{d=1}^{|O|} p^{d-1}\frac{|O_d|}{d}, \tag{4.5}$$

where O is the set of overlapping elements, $|O_d|$ counts the number of overlaps out of the first d elements, and p is a parameter governing how steep the decline in

Table 4.15: Results obtained by experimenting on the WiC dataset. Figures report the accuracy obtained for the three portions of the dataset and divided by POS.

System	Test	Training			Development		
		All	Nouns	Verbs	All	Nouns	Verbs
Contextualised word embeddings							
BERT-large[1]	68.4	-	-	-	-	-	-
WSD[2]	67.7	-	-	-	-	-	-
Ensemble[3]	66.7	-	-	-	-	-	
BERT-large[4]	65.5	-	-	-	-	-	-
ELMo-weighted[5]	61.2	-	-	-	-	-	-
Context2vec[4]	59.3	-	-	-	-	-	-
Elmo[4]	57.7	-	-	-	-	-	-
Sense representations							
DeConf[4]	58.7	-	-	-	-	-	-
SW2V[4]	58.1	-	-	-	-	-	-
JBT[4]	53.6	-	-	-	-	-	
LESSLEX (c-rnk-sim)	58.9	59.4	58.8	60.1	60.5	58.0	64.6
LESSLEX (RBO)	59.2	61.1	59.4	62.9	63.0	62.0	64.6
N2V (c-rnk-sim)	-	-	54.1	-	-	53.2	-
N2V (RBO)	-	-	60.7	-	-	63.4	-

[1] Wang et al. (2019)
[2] Loureiro and Jorge (2019b)
[3] Soler, Apidianaki, and Allauzen (2019)
[4] Mancini et al. (2017)
[5] Ansell, Bravo-Marquez, and Pfahringer (2019)

weights is: setting p to 0 would imply considering only the top element of the rank. In this setting, a low RBO score can be interpreted as indicating that senses that are closest to the contexts are different (thus suggesting that the sense intended by the polysemous term is different across texts), whilst the opposite case indicates that the senses more fitting to both contexts are same or similar, thereby authorizing to judge them as similar. For the task at hand, we simply assigned same sense when the RBO score exceeded a threshold set to 0.8.[15]

Results The results obtained experimenting on the WiC dataset are reported in Table 4.15.

Previous results show that this dataset is very challenging for embeddings that

[15]The RBO parameter p has been optimized and set to .9, which is a setting also in accord with literature (Webber et al., 2010).

do not directly grasp contextual information. The results of systems participating to this task can then been arranged into three main classes: those adopting embeddings featured by contextualised word embeddings, those experimenting with embeddings endowed with sense representations, and those implementing sentence level baselines (Pilehvar & Camacho-Collados, 2019). Given that the dataset is balanced (that is, it comprises an equal number of cases where the meaning of the polysemous term is preserved/different across sentences), and the fact that the task is a binary classification one, the random baseline is 50% accuracy. Systems employing sense representations (directly comparing to ours) obtained up to 58.7% accuracy score (Pilehvar & Collier, 2016). On the other side, those employing contextualized word embeddings achieved accuracy ranging from 57.7% accuracy (ELMo 1024-d, from the first LSTM hidden state) to 68.4% accuracy (BERT 1024-d, 24 layers, 340M parameters) (Pilehvar & Camacho-Collados, 2019).

Our resource directly compares with multi-prototype, sense-oriented, embeddings, namely JBT (Pelevina, Arefiev, Biemann, & Panchenko, 2016), DeConf (Pilehvar & Collier, 2016), and SW2V (Mancini et al., 2017). In spite of the simplicity of both adopted approaches (c-rnk-sim and RBO), by employing LESSLEX vectors we obtained higher accuracy values than those reported for such comparable resources (listed as 'Sense representations' in Figure 4.15).

We also experimented with N2V (with both c-rank-sim and RBO metrics), whose results are reported for nouns on the training and development subsets.[16] For such partial results we found slightly higher accuracy than obtained with LESSLEX with the RBO metrics. Unfortunately, however, N2V results can be hardly compared to ours, since the experiments on the test-set were executed through the CodaLab Competitions framework.[17] In fact the design of the competition does not permit to separate the results for nouns and verbs, as the gold standard for the test set is not publicly available,[18] so that we were not able to directly experiment on the test-set to deepen comparisons.

[16] Parameters setting for NASARI2VEC: in the c-rnk-sim, α was set to 0.7, and the threshold to 0.8; in the RBO run, p was set to 0.9 and the threshold to 0.9.
[17] https://competitions.codalab.org/competitions/20010.
[18] As of mid August 2019.

4.2.3 Semantic Text Similarity Task

As our third and final evaluation we consider the *Semantic Text Similarity* (STS) task, an extrinsic task that consists in computing a similarity score between two given portions of text. STS plays an important role in a plethora of applications such as information retrieval, text classification, question answering, topic detection, and as such it is helpful to evaluate to what extent LESSLEX vectors are suited to a downstream application.

Experimental setup We provide our results on two datasets popular for this task: the STS benchmark, and the SemEval-2017 Task 1 dataset, both by Cer, Diab, Agirre, Lopez-Gazpio, and Specia (2017). The former dataset has been built by starting from the corpus of English SemEval STS shared task data (2012-2017). Sentence pairs in the SemEval-2017 dataset feature a varied cross-lingual and multilingual setting, deriving from the Stanford Natural Language for Inference (SNLI) (Bowman, Angeli, Potts, & Manning, 2015) except for one track (one of two Spanish-English cross-lingual tasks, referred to as Track 4b. spa-spa), whose linguistic material has been taken from the WMT 2014 quality estimation task by Bojar et al. (2014). The translations in this dataset are the following: Arabic (ara-ara), Arabic-English (ara-eng), Spanish (spa-spa), Spanish-English (spa-eng), Spanish-English (spa-eng), English (eng-eng), Turkish-English (tur-eng).

To assess our embeddings in this task, we used the implementation of the HCTI system, participating in the SemEval-2017 Task 1 (Shao, 2017), kindly made available by the author.[19] HCTI obtained the overall third place in that SemEval competition. The HCTI system —implemented by using Keras (Chollet et al., 2015) and Tensorflow (Abadi et al., 2016)—generates sentence embeddings with twin convolutional neural networks (CNNs); these are then compared through the cosine similarity metrics, and element-wise difference with the resulting values is fed to additional layers to predict similarity labels. Namely, a Fully Connected Neural Network (FCNN) is used to transfer the semantic difference vector to a probability distribution over similarity scores. Two layers are employed herein, the first one using 300 units with *tanh* activation function; the second layer is charged to

[19] http://tiny.cc/dstsaz.

compute the (similarity label) probability distribution with 6 units combined with *softmax* activation function. While the original HCTI system employs GloVe vectors (Pennington et al., 2014), we used LESSLEX vectors in our experimentation.

In order to actually compare only the employed vectors by leaving unaltered the rest of the HCTI system, we adopted the same parameter setting as available in the software bundle implementing the approach proposed in (Shao, 2017). We were basically able to reproduce the results of the paper, except for the hand-crafted features; however, based on experimental evidence, these did not seem to produce significant improvements in the system's accuracy.

We devised two simple strategies to choose the word-senses to be actually fed to the HCTI system. In the first case we built the context vector (as illustrated in Equation 4.3), and selected for each input term the sense closest to such vector. The same procedure has been run on both texts being compared for similarity. In the following we refer to this strategy as to *c-rank*. In the second case we selected for each input term the sense closest to the terminological vector, in the same spirit as in the first component of the ranked similarity (rnk-sim, Equation 4.2). In the following this strategy is referred to as *t-rank*.

As mentioned, in the original experimentation two runs of the HCTI system were performed: one exploiting MT to translate all sentences into English, and another one with no MT, but performing a specific training on each track, depending on the involved languages (Shao, 2017, p.132). Since we are primarily interested in comparing LESSLEX and GloVe vectors, rather than the quality of services for MT, we experimented in the condition with no MT. However, in this setting the GloVe vectors could not be directly used to deal with the cross-lingual tracks of the SemEval-2017 dataset. Specific retraining (although with no handcrafted features) was performed by the HCTI system using the GloVe vectors on the multilingual tracks. In experimenting with LESSLEX vectors, the HCTI system was trained only on the English STS benchmark dataset also to deal with the SemEval-2017 dataset: that is, no Machine Translation step nor any specific re-training was performed in experiments with LESSLEX vectors to deal with cross-lingual tracks.

Results Results are reported in Table 4.16, where the correlation scores obtained by experimenting with LESSLEX and GloVe vectors are compared.

Table 4.16: Results on the STS task. Top: results on the STS benchmark. Bottom: results on the SemEval-2017 dataset. Reported results are Pearson correlation indices, measuring the agreement with human annotated data. In particular, we compare the Pearson scores obtained by the HCTI system using LESSLEX and GloVe vectors. As regards as the runs with GloVe vectors, we report results with no hand-crafted features (no HF), and without machine translation (no MT)

STS Benchmark (English)

Track	HCTI + LESSLEX		HCTI + GloVe
	(t-rank)	(c-rank)	(no HF)
dev	.819	.823	.824
test	.772	.786	.783

SemEval 2017

Track	HCTI + LESSLEX		HCTI + GloVe
	(t-rank)	(c-rank)	(no MT)
1. ara-ara	.534	**.618**	.437
2. ara-eng	.310	.476	-
3. spa-spa	**.800**	.730	.671
4a. spa-eng	.576	.558	-
4b. spa-eng	.143	.009	-
5. eng-eng	.811	.708	**.816**
6. tur-eng	.400	.433	-

Let us start by considering the results obtained by experimenting on the STS benchmark. Here, when using LESSLEX embeddings we obtained figures similar to those obtained by the HCTI system using GLoVe vectors; namely, we observe that the choice of senses based on the overall context (c-rank) provides little improvements with respect to both GloVe vectors and to the t-rank strategy.

As regards as the seven tracks in the SemEval-2017 dataset, we can distinguish between results on multilingual and cross-lingual subsets of data. As regards as the former ones (that is, the ara-ara, spa-spa and eng-eng tracks), HCTI with LESSLEX obtained higher correlation scores than when using GloVe embeddings in two cases: $+0.181$ on the Arabic task, $+0.129$ on the Spanish task, and comparable results (-0.005) on the English track. We stress that no re-training was performed on LESSLEX vectors on languages different from English, so that the improvement obtained in the tracks 1 and 3 (ara-ara and spa-spa, respectively) is even more relevant. We interpret this achievement as stemming from the fact that LESSLEX vec-

tors contain both conceptual and terminological descriptions: this seems also to explain the fact that the advantage obtained by employing LESSLEX vectors w.r.t. GloVe is more sensible for languages where the translation and/or re-training are less effective, such as pairs involving either the Arabic or Turkish language. Also, we note that using contextual information (c-rank strategy) to govern the selection of senses ensures comparable results to the t-rank strategy across settings (with the exception of track 4b, where the drop in the correlation is very prominent, in one order of magnitude). Finally, it is interesting to observe that in dealing with cross-lingual texts that involve arguably less-covered languages (i.e., in the tracks 2 and 6, ara-eng and tur-eng), the c-rank strategy produced better results than the t-rank strategy.

To summarize the results on the STS task, by plugging LESSLEX embeddings into a state-of-the-art system such as HCTI we obtained results that either improve or are comparable to more computationally intensive approaches involving either MT or re-training, necessary to use GLoVe vectors in a multilingual and cross-lingual setting. One distinguishing feature of our approach is that of hosting terminological and conceptual information in the same semantic space: experimental evidence seems to confirm it as helpful in reducing the need for further processing, and beneficial to map different languages onto such unified semantic space.

4.2.4 General Discussion

Our experimentation has taken into account overall eleven languages, from different linguistic lineages, such as Arabic, coming from the Semitic phylum; Basque, a language isolate (reminiscent of the languages spoken in southwestern Europe before Latin); English and German, two West Germanic languages; Farsi, that as an Indo-Iranian language can be ascribed to the set of Indo-European languages; Spanish and Portuguese, that are Western Romance languages in the Iberian-Romance branch; French, from the Gallo-Romance branch of Western Romance languages; Italian, also from the Romance lineage; Russian, from the eastern branch of the Slavic family of languages; Turkish, in the group of Altaic languages, featured by phenomena such as vowel harmony and agglutination.

We employed LESSLEX embeddings in order to cope with three tasks: *i)* the tra-

ditional semantic similarity task, where we experimented on six different datasets (RG-65, WS-Sim-353, SimLex-999, SimVerbs-3500, SemEval-2017 (Task 2) and Goikoetxea-2018); *ii)* the contextual semantic similarity task, where we experimented on two datasets, SCWS and WiC; *iii)* the STS task, where the STS Benchmark and the SemEval-2017 (Task 1) dataset were used for the experimentation.

In the first mentioned task (Section 4.2.1) our experiments show that in most cases LESSLEX results improve on those by all other competitors. As competitors all the principal embeddings were selected that allow coping with multilingual tasks: ConceptNet Numberbatch, NASARI, JOINTCHYB, SenseEmbed, and Nasari2Vec. Two different experimental conditions were considered (MSV and CbA, Table 4.11). Both views on results indicate that our approach outperforms the existing ones. To the best of our knowledge this is the most extensive experimentation ever performed on as many benchmarks, and including results for as many resources.

In dealing with the Contextual Similarity task (Section 4.2.2) we compared our results with those obtained by using NASARI2VEC, which also contains descriptions for both terms and nominal concepts in the same semantic space, and with results available in literature. The obtained figures show that despite not being tuned for this task, our approach improves on previous results on the SCWS dataset. On the WiC dataset, results obtained by experimenting with LESSLEX vectors overcome all those provided by directly comparable resources. Results obtained by state-of-the-art approaches (employing contextualized sense embeddings) in this task are about 9% above those currently achieved through sense embeddings.

As regards as the third task on Semantic Text Similarity (Section 4.2.3), we used our embeddings by feeding them to a Convolutional Neural Network in place of GloVe embeddings. The main outcome of this experiment is that while our results are comparable to those obtained by using GloVe for English tracks, they improve on the results obtained with GloVe in the cross-lingual setting, even though these are specifically retrained on the considered tracks.

In general, handling sense-embeddings involves some further processing to select senses for input terms, while with word-embeddings one can typically benefit from the direct mapping term-vector. Hence, the strategy employed to select senses is relevant when using LESSLEX embeddings. Also — though indirectly — sub-

ject to evaluation was the proposed similarity metrics of ranked-similarity; it basically relies on ranking sense vectors based on their distance from the terminological one. Ranked-similarity clearly outperforms the maximization of cosine similarity on LESSLEX embeddings. Besides, the contextual ranked-similarity (which was devised to deal with the contextual similarity task) showed to perform well, by taking into account information from the context vector rather than from the terminological one. We defer to further work an exhaustive exploration of their underlying assumptions and the analytical description of differences in computing conceptual similarity between such variants of ranked similarity and existing metrics such as, e.g., the Rank-Biased Overlap.

5 Sense Identification

In this chapter we introduce a line of research closely related to semantic similarity, concerned with *sense identification* (Colla, Mensa, & Radicioni, 2020b). More specifically, we argue that semantic similarity needs to be complemented by another task, involving the identification of the senses at the base of the similarity rating. Dealing with such novel task involves refining the standard cosine-maximization approach that has traditionally featured the semantic similarity task.

The Chapter is organized as follows: we start by reporting on the role of semantic similarity in knowledge representation and NLP fields (Section 5.1). In Section 5.2 we introduce two novel semantic similarity metrics that have been devised to deal with both semantic similarity and sense identification, namely the ranked similarity (Section 5.2.1) and the semantic neighborhood similarity (Section 5.2.2). Afterwards, in Section 5.3 we report on the evaluation of such metrics. More precisely, we first illustrate the SemEval-2017 dataset (Section 5.3.1), then we report on the annotation procedure so as to build the dataset for the sense identification task (Section 5.3.2) together with the employed resources (Section 5.3.3) and the sense retrieval strategy (Section 5.3.4). Finally, we present and discuss the results of the assessed resources on the sense identification task (Sections 5.3.5 and 5.3.6, respectively).

5.1 Introduction

Similarity plays a fundamental role in theories of knowledge and behavior. It serves as an organizing principle by which individuals "classify objects, form concepts, and make generalizations" (Tversky, 1977). It is central in models investigating many sorts of cognitive processes, such as categorization (Goldstone, 1994), problem solving (Novick, 1988), analogical reasoning (Gentner & Smith, 2012), and

it underlies many forms of automatic reasoning, such as case-based and similarity-based reasoning (Lamperti & Zanella, 2006; Sun, 1995). In the field of Computational Linguistics, similarity is relevant to various tasks, in particular in the computation of *semantic similarity*.

Semantic similarity is a long-standing topic of investigation (see, e.g., (Baddeley, 1966a, 1966b; Schaeffer & Wallace, 1969)), but it is in the last few years that it has emerged as a central one: historically, this phenomenon is related to various aspects, such as the growing needs for elaborating natural language at large, and the wide availability of high quality word embeddings. Semantic similarity can be addressed at different linguistic levels, such as the sense level (or word *meaning*), and the term level (or word *form*) (Pilehvar & Navigli, 2015). In the former case the input is composed of a sense pair, identified based on some sense inventory, such as WordNet (G. A. Miller, 1995) or BabelNet (Navigli & Ponzetto, 2010). However, although many approaches have been proposed to cope with various sorts of information at the term level, sense level and mixed $\langle term, sense \rangle$ level, existing methods to compute semantic similarity mostly rely on maximizing the cosine similarity between vector pairs. This chapter addresses two distinct though related research questions. We start from sense embeddings that map sense and term descriptions onto a shared semantic space. Our first research question is whether and to what extent such representations can be exploited in building novel and most accurate metrics to compute semantic similarity. The second research question is whether the semantic similarity task can be paired to another task aimed at identifying which senses are actually involved in the semantic similarity rating. In other words, we posit that sense identification is a natural and crucial complement to the semantic similarity. In the attempt at answering both research questions, we elaborate on principles that are largely acknowledged to contribute to human lexical competence, and propose two novel similarity metrics that allow addressing both questions in a unified fashion. To test our hypothesis on the sense identification task, we annotated with senses the English version of the SemEval-2017 Task 2 dataset (Camacho-Collados et al., 2017), composed by 500 word pairs and their similarity scores. As far as we know this is the first dataset for semantic similarity that has been also annotated with sense identifiers. The resulting dataset is featured by

high inter-rater agreement score, thereby providing a reliable experimental base. To evaluate the novel similarity metrics we selected six recently proposed sense embeddings, and used them *i)* to compute the semantic similarity featuring pairs in the mentioned dataset, and *ii)* to identify the involved senses. The obtained results show that the proposed metrics favorably compare to the familiar cosine similarity maximization, both in the semantic similarity task, and in the sense individuation task: our novel metrics can be simply plugged into existing systems to replace the maximization strategy. Finally, we analytically discuss the details of the considered resources and how these affect the novel metrics to provide valuable insights for applications and systems using sense embeddings.

5.2 Novel Semantic Similarity Metrics

Our proposal for novel similarity metrics is connected to the availability of different sense representations for each term, possibly (though not necessarily) linked to a sense inventory. Additionally, the semantic similarity task should involve also the sense identification, such that the similarity score is linked to a sense pair. As illustrated in what follows, this step may be not only relevant to fully assess similarity scores computed by computer systems, but also beneficial to improve the correlation with human rating. Let us start by recalling the aforementioned formula to compute the semantic similarity for a term pair $\langle t, u\rangle$ as the maximal cosine similarity ($\mathcal{M}$-sim) featuring all sense combinations $\langle s^t, s^u\rangle$,

$$\mathcal{M}\text{-sim}(t,u) = \max_{\substack{s^t \in S^t,\\ s^u \in S^u}} \Big(\text{cos-sim}(s^t, s^u)\Big). \tag{5.1}$$

In this setting, retrieving the sense pair $\langle s^t, s^u\rangle$ that underlie the maximal similarity score amounts to finding the sense pair that maximizes the above expression, that is

$$\langle s^t, s^u\rangle \leftarrow \arg\max_{\substack{s^t \in S^t,\\ s^u \in S^u}} \Big(\text{cos-sim}(s^t,\ s^u)\Big). \tag{5.2}$$

Table 5.1: List of senses associated to the terms Weather and Wave in BabelNet.

Synset ID & Title (Weather)	**Description**
bn:00080759n; weather forecast; ...	A forecast of the weather
bn:14903090n; Weather (album)	Weather is the 9th studio album by ...
bn:00006808n; atmospheric condition; ...	The atmospheric conditions...
Synset ID & Title (Wave)	**Description**
bn:17074220n; Wave (CNBLUE album)	Wave is the third studio album by ...
bn:00080690n; wave	A persistent and unusual weather condition...
bn:00056171n; moving ridge; ...	Ridges that move across the surface of a liquid...
bn:00079034n; undulation; wave	An undulating curve
bn:00079036n; wave; undulation	(physics) a movement up and down...
bn:00080687n; wave	A movement like that of a sudden occurrence...
bn:00080688n; wave	Something that rises rapidly
bn:00080689n; wave	Undulations in the hair
bn:00041739n; greeting; Wave (Social)	Usually plural expression of good will...
bn:02765490n; WAV; WAVE; .wav	Waveform Audio File Format...
bn:13892600n; wave (gesture)	Nonverbal communication gesture...
bn:14555074n; Wave (A. C. Jobim song)	Bossa nova song...

EXAMPLE As our working example, let us consider the word pair ⟨Weather,-Wave⟩—picked from the SemEval2017 Task 2 dataset (Camacho-Collados et al., 2017)—; also, we will use LESSLEX sense embeddings (Colla, Mensa, & Radicioni, 2020a). We target the BabelNet sense inventory, where 3 senses can be found for Weather, and 38 for Wave, some of which are presented in Table 5.1.[1] All 114 resulting combinations will thus be inspected in order to compute the maximization involved in the computation of the semantic similarity. In Table 5.2 we present the scores obtained for four sense pairs; in particular, the table shows the maximal score, and the associated pair $\langle s^t, s^u \rangle$: such top scoring sense pair is ⟨bn:14903090n, bn:17074220n⟩, where the former sense refers to an album by an American singer for Weather, and the latter refers to an album by a South Korean rock band for Wave. Other combinations containing the same musical album for Weather involve a song by the Brazilian composer Jobim; the wave movement;[2] and the .wav file format. Putting together two musical albums, an album and a song, or a musical entity and an audio file format is intuitively reasonable, since

[1] Full sense lists can be retrieved at the URLs https://babelnet.org/search?word=weather&lang=EN and https://babelnet.org/search?word=wave&lang=EN, respectively.

[2] Like in 'Troops advancing in waves', WordNet usage example.

Table 5.2: Cosine similarity scores computed by employing LESSLEX vectors employing the sense associated to the 'Weather (album)' (bn:14903090n) and different senses for the term Wave. The top similarity score is marked by bold font.

Sense ID (Weather)	Sense ID and description (Wave)	cos-sim score
bn:14903090n	bn:17074220n (Wave (CNBLUE album))	**0.6062**
bn:14903090n	bn:14555074n (Wave, A. C. Jobim song)	0.3713
bn:14903090n	bn:00080687n (Wave movement)	0.2470
bn:14903090n	bn:02765490n (WAV; WAVE; .wav)	0.2070

these senses are to some extent connected. However, humans requested to rate the similarity and to indicate the senses underlying the similarity score would rather indicate senses dealing with atmospheric conditions or forecasts, and senses dealing with some kind of physical waves.[3] Such senses are more central, while the entities associated to music are less relevant in a general setting. Conversely, the rationale implemented by the maximization considers only the closeness of senses: based on these accounts, the mentioned music albums obtain maximal score (due to the fact that their vectorial representations are closer than weather forecasts/conditions and whatever physical wave). This fact is graphically illustrated in Figure 5.1.

This example shows that the closest senses exhibit some commonalities, but they do not necessarily agree with human judgment. If we admit that the top scoring pair may not not be associated with the senses that are intended by human annotators, then the similarity score is immediately undermined, as a property featuring the possibly wrong (better, differing from human response) sense pair. Hence the question is: Which similarity score should be returned instead? Different strategies and metrics can be envisaged, to identify the senses underlying the similarity score, along with the score itself: in the rest of the Section we introduce our proposal to overcome such limitations.

5.2.1 Ranked Similarity

In order to identify the senses activated by the similarity judgments, we devised a mechanism aimed at ranking senses inspired by a popular notion in Cognitive Science, *availability*. Availability is in a pool of heuristics that bias human judg-

[3] The annotated dataset devised for the present work and the adopted annotation methodology are illustrated in Section 5.3.2.

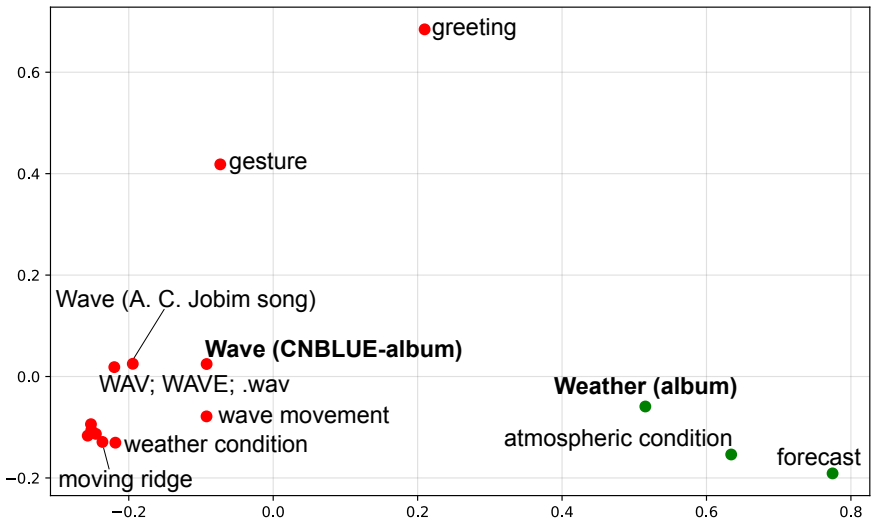

Figure 5.1: Plot of some of the senses associated to Wave (in red) and to Weather (in green). For the sake of readability, we did not print all labels for Wave senses. Senses marked with bold fonts indicate the top scoring senses associated to the CNBLUE album for Wave, and to the Weather album for Weather. The 300-dimensional LESSLEX embeddings (Colla, Mensa, & Radicioni, 2020a) were mapped onto two dimensions through the implementation of multidimensional scaling provided by scikit-learn (Pedregosa et al., 2011).

ment and action under uncertainty (Tversky & Kahneman, 1973). Availability has primarily to do with frequency, but it is affected by other factors, as well; in an operational definition, availability is a principle whereby "instances of large classes are recalled better and faster than instances of less frequent classes; [...] likely occurrences are easier to imagine than unlikely ones; [...] the associative connections between events are strengthened when events frequently co-occur" (Tversky & Kahneman, 1974, p.1128). We thus hypothesize that similar mechanisms govern lexical access and semantic choice in the setting of semantic similarity: the relevance of a given sense (with respect to the central meaning associated to the term) should be used to refine the closest-senses heuristics implemented by the max-similarity approach.

In order to individuate the senses from the term pair, we have devised the *ranked-similarity* metrics, $\mathcal{R}$-sim. Ranked similarity refines the aforementioned max-similarity approach by also taking into account how central a sense is to the considered term. Given a term t, we impose a total ordering on its senses based on the

distance intervening between each sense vector and the term vector. In formulae,

$$\sqsupseteq_t \equiv \{(s_x^t, s_y^t) \mid \text{if cos-sim}(s_x^t, t) \geqslant \text{cos-sim}(s_y^t, t)\} \tag{5.3}$$

a generic sense s_x^t is more relevant than s_y^t with respect to the term representation of t if $s_x^t \sqsupseteq_t s_y^t$. On this base we compute the *ranking* for all senses associated to t, from the most relevant to the less relevant, in such a way that for any i-th sense of t we are able to retrieve the rank(s_i^t) as its index in the $\sqsupseteq_t$ ordering. Such ranking can be computed in all sense embeddings that map both terminological and conceptual representations onto a shared semantic space, which is rather common in recent sense embeddings such as, e.g., (Camacho-Collados et al., 2015b; Colla, Mensa, & Radicioni, 2020a; Iacobacci & Navigli, 2019; Iacobacci et al., 2015; Mancini et al., 2017; Pilehvar & Collier, 2016). Earlier in this section we introduced an example in which sense identification was misled by closest senses, that may override the senses actually considered by humans. In order to overcome this issue, $\mathcal{R}$-sim combines the relevance of each sense and the classical maximization approach. Given two terms t and u and their respective sets of senses $S^t = \{s_1^t, \ldots, s_n^t\}$, $S^u = \{s_1^u, \ldots, s_k^u\}$, we define the ranked-similarity score as

$$\mathcal{R}\text{-sim}(s_i^t, s_j^u) = \left[(1-\alpha) \cdot \Big(\text{rank}(s_i^t) + \text{rank}(s_j^u)\Big)^{-1} + \Big(\alpha \cdot \text{cos-sim}(s_i^t, s_j^u)\Big)\right], \tag{5.4}$$

where rank(s_i^t) and rank(s_j^u) are the rank of the sense vector with respect to the term vectors associated to t and u, respectively. The α factor is used to tune the balance between ranking factor and raw cosine similarity, and it varies in the interval $(0, 1)$.[4] Rank is thus used to emphasize the relevance of senses whose vector representation is closer to that of the term vector conflating all senses for the given term: in this way, the contribution of each sense to the overall score decreases as its vector is less related to the terminological one. The functioning of the $\mathcal{R}$-sim metrics is graphically illustrated in Figure 5.2, where also the term vectors representing Weather and Wave are plotted. This metrics can be plugged into the general formula reported in Equation 5.1 in place of cosine similarity.

Likewise, $\mathcal{R}$-sim is used to replace cosine similarity to perform sense identifica-

[4] Actual values assigned to α while experimenting with different resources are reported in Table 5.4.

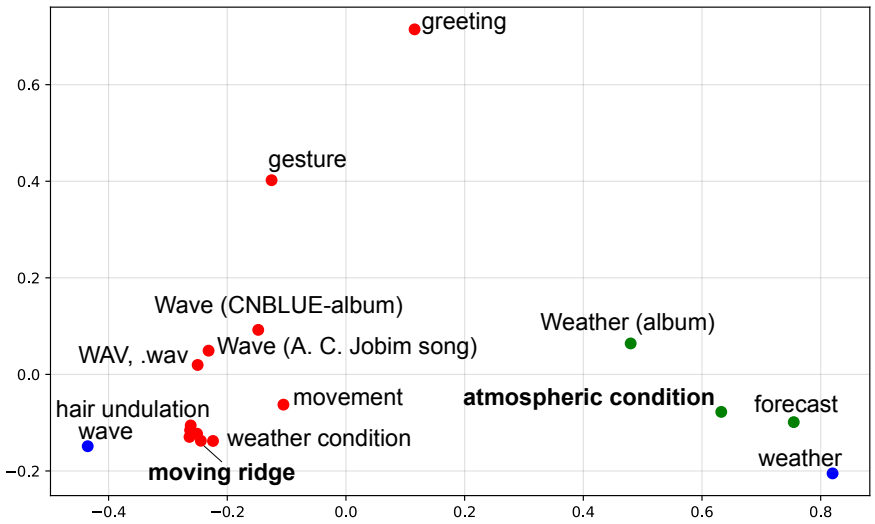

Figure 5.2: Graphical illustration of the working rationale of the $\mathcal{R}$-sim metrics, combining both distance between sense and term vector, and distance between sense pair. Red and green dots represent sense representations, while blue dots correspond to term representations, conflating all senses for each term. The pivot senses individuated through the $\mathcal{R}$-sim metrics are marked with bold fonts.

tion (please refer to Equation 5.2). In fact, the ranked-similarity also implements a criterion to choose two senses from S^t and S^u. In this setting, the sense identification basically involves the choice of the arguments maximizing the $\mathcal{R}$-sim score by replacing the cosine-similarity in Equation 5.2, in such a way that

$$\langle \hat{s}_i^t, \hat{s}_j^u \rangle \leftarrow \underset{s_i^t \in S^t,\, s_j^u \in S^u}{\arg\max}\ \mathcal{R}\text{-sim}(s_i^t, s_j^u). \tag{5.5}$$

The senses $\hat{s}_i^t$ and $\hat{s}_j^u$, individuated through the $\mathcal{R}$-sim metrics, are referred to as *pivot* senses.

EXAMPLE We carry on with the previous example, and compute the $\mathcal{R}$-sim score for all sense pairs associated to the term pair Weather and Wave; a set of the highest scoring sense pairs is provided in Table 5.3. While we formerly (based on cosine-similarity) picked the sense pair composed of two music albums, we now individuate a different sense pair, that is $\langle$bn:00006808n, bn:00056171n$\rangle$. The former sense refers herein to Weather intended as atmospheric condition, and the latter to Wave as ridge moving on the surface of a liquid. Interestingly, such sense pair is

Table 5.3: $\mathcal{R}$-sim scores computed by employing LESSLEX vectors for the pair ⟨Weather,Wave⟩. We present pairs including the sense associated to 'Weather (atmospheric condition)' (bn:00006808n), which is combined with varying senses for the term Wave; such senses can be grouped into the clusters of meaning illustrated in the column C. The top similarity score is marked by bold font.

Sense ID (Weather)	Sense ID and description (Wave)	C	$\mathcal{R}$-sim score
bn:00006808n	bn:00056171n (Moving ridge)	Physical	**0.398**
bn:00006808n	bn:00079036n (Wave, undulation (physics))		0.284
bn:00006808n	bn:00079034n (Undulation, wave)		0.242
bn:00006808n	bn:00080689n (Hair undulation)		0.200
bn:00006808n	bn:00080690n (Wave (unusual weather))		0.200
bn:00006808n	bn:00080687n (Wave movement)		0.177
bn:00006808n	bn:00080688n (Wave rising)		0.168
bn:00006808n	bn:02765490n (WAV; WAVE; .wav)	Music	0.105
bn:00006808n	bn:14555074n (Wave, A. C. Jobim song)		0.101
bn:00006808n	bn:17074220n (Wave (CNBLUE album))		0.074
bn:00006808n	bn:13892600n (Wave (gesture))	Greeting	0.047
bn:00006808n	bn:00041739n (Greeting; Wave (Social))		0.029

the same as annotated by humans, and the similarity score computed through the $\mathcal{R}$-sim (0.398 *vs.* a 0.207 ground truth score) is significantly closer to human rating, as well. The vector representations associated to the competing senses and to the term level description are portrayed in Figure 5.2. Furthermore, by observing both Figure 5.2 and Table 5.3 we find that senses listed for Wave can be grouped into few broader meaning classes, such as i) physical senses, ii) music-related senses (music albums, songs, compressed file format), and iii) greetings ('hand gesture' and 'expression of good will, especially on meeting'). In the next Section we introduce the approach devised to automatically detect and cluster senses that fall into a small semantic neighborhood, and the associated similarity metrics.

5.2.2 Semantic Neighborhood Similarity

Experimental evidence and theories from Cognitive Science exist suggesting that some senses from fine-grained sense inventories can be be grouped. Word senses are typically thought of as mutually disjoint: e.g., in the task of Word Sense Disambiguation it is common to assume that a single correct label exists for every markable. However, these assumptions have been challenged from many diverse perspectives, as reported by (Erk, McCarthy, & Gaylord, 2013): word meaning can be described as having fuzzy boundaries, and semantic annotation and judgment admit graded membership. Moreover, at least in some cases, it is known that word

meaning is not separable into senses distinct enough to guarantee consistent annotation (J. Chen & Palmer, 2009). Another observation supports the plausibility of sense groupings: even senses that appear clearly separated are actually hard to distinguish in certain contexts, as illustrated by Erk et al. (2013). In another view, theories have been proposed that suggest to interpret polysemy as *sense modulation*, through either specialization or broadening of meaning in context (Copestake & Briscoe, 1995). In this view, two forms of sense modulation have been explored, named constructional polysemy (dealing with sense modulation) and sense extensions (dealing with sense change). In either case, reconstructing the main unities of sense by possibly merging close senses seems to be compatible with such meaning modulation devices. Different arguments have been proposed to support the hypothesis that different levels of granularity in the sense inventory might be appropriate for different tasks, and in general it is largely acknowledged that the fine-grained sense distinctions may be detrimental to various NLP tasks (Mihalcea & Moldovan, 2001; Resnik & Yarowsky, 1999; Tomuro, 2001). While different sources of evidence exist in favor of separating senses, no general consensus has been reached on which criteria should be actually followed (E. Agirre & De Lacalle, 2003). For example, a form of underspecification has been proposed to deal with the word sense disambiguation (WSD) task (Buitelaar, 2000). In this view, for some sorts of applications, such as text categorization and information extraction, more coarse-grained sense inventories are preferable, while fine-grained sense distinctions are necessary for precise tasks such as machine translation (Ng, Wang, & Chan, 2003). Moreover, different levels of sense granularity have been explored, such as PropBank Framesets, WordNet sense groupings, and an additional intermediate level of granularity (Palmer, Babko-Malaya, & Dang, 2004). Further work concentrated on the topic of clustering senses (Lieto, Mensa, & Radicioni, 2016b; Navigli, 2006), to tame the sense sparsity menacing the WSD task. Finally, research explicitly targeting annotation procedures, and specifically concerned with the preparation of sense-tagged corpora for SemEval tasks reports that most annotators disagreements were detected between "closely related WordNet senses with only subtle (and often inexplicit) distinctions", thus demanding for more coarse-grained sense distinctions (Snyder & Palmer, 2004, p.42).

A different notion of similarity is rooted in the tradition of spatial cognition, and has to do with the idea of neighborhood, which is central for our present concerns. In this approach conceptual neighborhood is used as a tool for conceptual categorization (Bouraoui, Camacho-Collados, Espinosa-Anke, & Schockaert, 2019), and an analogous categorization principle, based on a hybrid representational approach bringing together conceptual spaces and formal ontologies has been designed in (Lieto, Radicioni, & Rho, 2017).

To sum up, cited literature seems to suggest that senses should not be conceived as fully separate, static and pointwise entities, but rather as units of meaning partially and dynamically overlapping, also based on their contextual nature. Accordingly, we propose that in cases where slightly different, close —and less separated— senses can be individuated, a 'sense span' can be built as a conflation of nearby senses. Such features are hypothesized to work both for the semantic similarity and the sense individuation tasks, thereby allowing postulating a unified treatment to deal with the two connected tasks.

Based on such underpinnings, we devised a strategy to build vectors embodying the semantic neighborhood surrounding specific senses. However, different from the mentioned approaches aimed at clustering senses, we devised a strategy for the *on-line* grouping of senses, still preserving the fine-grained sense inventory provided by BabelNet: the underlying assumption is that senses are grouped dynamically, in a context-dependent fashion, which is compatible with the *sense modulation* proposed by (Copestake & Briscoe, 1995) and with the Generative Lexicon hypothesis (Pustejovsky, 1991).

We first give an intuition about the semantic neighborhood similarity, and then we illustrate it with full detail. We define the neighborhood-proximity metrics ($\mathcal{N}$-prox) as an instrument to evaluate the semantic closeness of lexical items by leveraging two main insights: *i)* to decide whether a given sense s_i^t is close enough to the pivot sense (as returned by the $\mathcal{R}$-sim metrics; please refer to Equation 5.5) to be admitted to the neighborhood, we compare it also to the term representation; in so doing, *ii)* the neighborhood-proximity is built in a dynamic fashion, by starting from the pivot sense $\hat{s}_i^t$. In order to build the semantic neighborhood, we start from the pivot sense pair $\langle \hat{s}^t, \hat{s}^u \rangle$, $\hat{s}^t \in S^t$ and $\hat{s}^u \in S^u$ (Equation 5.5). Pivot senses are

then used to build the sets $\hat{S}^t \subseteq S^t$ and $\hat{S}^u \subseteq S^u$, containing all *neighbor* senses, that are then averaged into synthetic vector descriptions. The two vectors built by starting from pivot senses are finally compared in order to obtain the similarity score, and the senses hosted in the semantic neighborhood are selected as those at the base of the similarity rating.

Given the vectorial representations for the term t, for the pivot sense $\hat{s}^t$ and for a specific sense s_i^t, the neighborhood-proximity metrics $\mathcal{N}$-prox is defined as follows:

$$\mathcal{N}\text{-prox}(t, \hat{s}^t, s_i^t) = \left[\beta \cdot \text{cos-sim}(\hat{s}^t, s_i^t)\right] + \left[\Big(1-\beta\Big) \cdot \Big(1 - \mid \text{cos-sim}(t, s_i^t) - \text{cos-sim}(t, \hat{s}^t) \mid \Big)\right]. \tag{5.6}$$

The $\mathcal{N}$-prox metrics takes into account the raw cosine similarity between the two sense vectors $\hat{s}^t$ and s_i^t combined with the absolute difference between their distance from the term vector t. In particular, the second term in Equation 5.6 is designed so to reward senses whose similarity with the term vector closely approaches the similarity featuring term and pivot sense vectors. Since the pivot sense has been individuated by maximizing its closeness to the term vector (and meantime to a sense for the other term), it follows that the heuristics overall implemented through the $\mathcal{N}$-prox metrics is aimed at over-weighting a sense s_i^t if it is similar to both t and $\hat{s}^t$, thus implementing a preference for more salient senses among those that are also close to the pivot. The parameter β varies in the interval $(0, 1)$, and is designed to govern the relative strength of the two components: traditional cosine similarity score, and preference for 'central' and salient senses. Setting β to 1 would thus amount to reverting to cosine similarity; at the opposite end, a β set to 0 would involve disregarding the cosine similarity component in favor of the heuristics defined by the second term of Equation 5.6.

The $\mathcal{N}$-prox is computed for each s_i^t and, if $\mathcal{N}$-prox yields a value that reaches the threshold γ, the sense is included in $\hat{S}^t$:

$$\hat{S}^t \leftarrow \underset{s_i^t \in S^t}{\text{for-each}}\Big(\text{select-if}(\mathcal{N}\text{-prox}(t, \hat{s}^t, s_i^t) \geq \gamma)\Big). \tag{5.7}$$

The parameter gamma γ was devised to tune the width of the resulting sense neighborhood. It varies in the interval $(0, 1)$: a γ approaching 1 implies a more restrictive

criterion to admit senses to the neighborhood of the pivot sense $\hat{s}_i^t$. Conversely, a γ set to 0 would imply a single wider neighborhood including all senses for t.[5]

We then build $\mathcal{N}$-vec^t, the centroid of all items in the semantic neighborhood by averaging all elements in $\hat{S}^t$:

$$\mathcal{N}\text{-vec}^t \leftarrow \text{avg}(s) \mid s \in \hat{S}^t. \tag{5.8}$$

The same procedure is applied to compute $\hat{S}^u$, the neighborhood of $\hat{s}^u$, finally attaining the centroid $\mathcal{N}$-vec^u. The semantic similarity between the two resulting vectors is then computed as

$$\mathcal{N}\text{-sim}(\mathcal{N}\text{-vec}^t, \mathcal{N}\text{-vec}^u) = \text{cos-sim}\Big(\mathcal{N}\text{-vec}^t,\ \mathcal{N}\text{-vec}^u\Big). \tag{5.9}$$

Similar to the $\mathcal{R}$-sim metrics, also the $\mathcal{N}$-sim metrics provides a criterion to select senses for the sense individuation task. In this case, we identify a pool of close (γ-proximal) senses that are included in $\hat{S}^t$ and $\hat{S}^u$. At evaluation time, we compute precision and recall of such sets.

EXAMPLE Let us consider how the $\mathcal{N}$-sim metrics applies to the term pair ⟨Weather,-Wave⟩. We start by individuating the pivot sense pair through the $\mathcal{R}$-sim, that is the pair ⟨bn:00006808n, bn:00056171n⟩. As we already know, the first sense refers to the atmospheric Weather condition, while the latter one denotes Wave as 'ridge moving across the surface of a liquid'. The sets $\hat{S}^t$ and $\hat{S}^u$ containing all *neighbor* senses are then computed through the $\mathcal{N}$-prox metrics, and the centroid for each neighborhood of senses is built based on Equation 5.8. The final step produces a synthetic representation of the whole semantic neighborhood that can be directly compared with other vectorial representations. The senses involved in the semantic neighborhood are graphically illustrated in Figure 5.3. By adjusting γ, it is possible to characterize the final neighborhoods with different degrees of specificity, according to the possibly varied needs for more or less coarsened representations. While targeting the conceptual similarity task, we explored a set of parameters so to ensure that only very close (γ-proximal) senses are added to the neighbors' set

[5] The actual values assigned to β and γ while experimenting with the different resources are reported in Table 5.4.

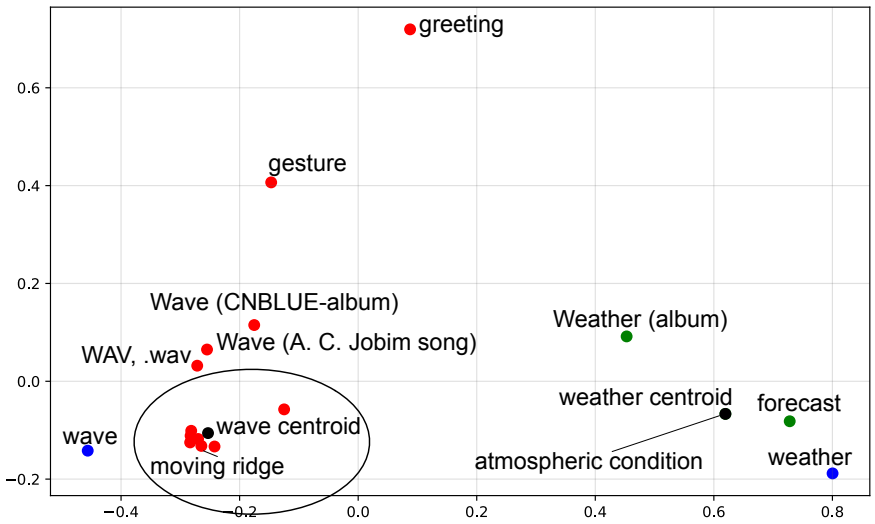

Figure 5.3: Graphical illustration of the working rationale of the $\mathcal{N}$-sim metrics, using the senses pair retrieved by employing the $\mathcal{R}$-sim and extending them with their neighbors. Red and green dots represent sense representations, blue dots correspond to term representations, conflating all senses for each term, while black dots represent the centroids of each semantic neighborhood. The oval shape shows the senses included in the neighborhood built around the pivot sense for Wave (that is, wave as 'moving ridge').

$\hat{S}^t$. For example, on the other side the senses for Weather are not close enough to be merged, thereby resulting in three singleton neighborhoods. Conversely, if we focus on the neighborhood built around the pivot sense of Wave intended as the 'moving ridge' sense (marked with the circle in Figure 5.3), we obtain the senses presented in the plot of Figure 5.4. Also, neighborhoods depend on the pivot sense: this fact determines that senses admitted to $\hat{S}^t$ —the neighborhood of the pivot sense— change in accordance to the second term (u) in the term pair $\langle t, u \rangle$. We have seen that the senses of the neighborhood of 'moving ridge' (in the context of the pair $\langle$Weather,Wave$\rangle$) are mostly concerned with physical movements. If we move to the term pair $\langle$Wave, Song$\rangle$, $\mathcal{R}$-sim returns the pivot senses bn:17141665n, associated to the song 'Wave (Beck song)', and bn:00072794n, referred to 'song, vocal, Voice type'. In this case, the senses collected for the neighborhood of Wave include other songs, such as a song by Jobim, and a song by Patti Smith. Interestingly, the $\mathcal{N}$-sim measure shows to be suited also to entities, and not only to conceptual representations.

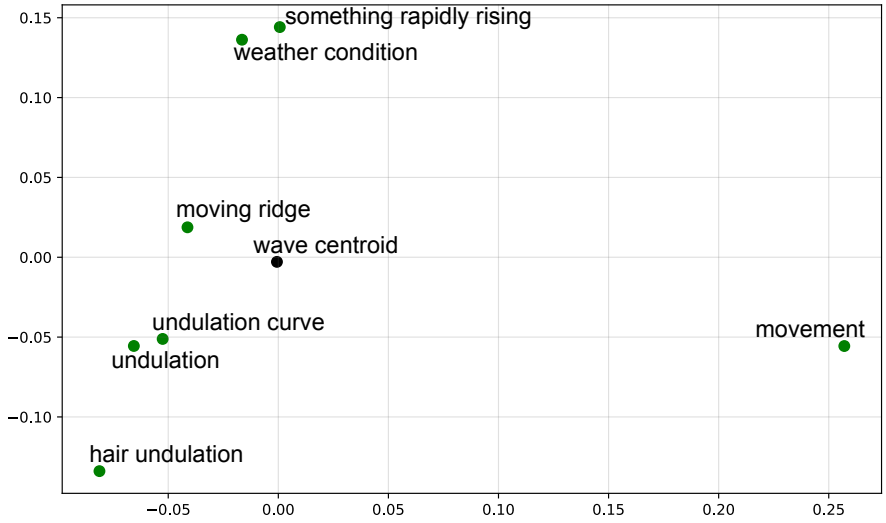

Figure 5.4: Graphical illustration of the semantic neighborhood for the Wave word senses by starting from bn:00056171n (Moving ridge). Green dots represent sense representations, while the black dot represents the centroid of the semantic neighborhood.

5.3 Evaluation

To evaluate the proposed similarity metrics we devised a twofold experimentation. We tested on the semantic similarity task and on the sense individuation task: in both experimental settings we compared the results obtained through the $\mathcal{R}$-sim and $\mathcal{N}$-sim metrics against those obtained by employing the $\mathcal{M}$-sim (maximization of cosine similarity) metrics.

This Section is structured as follows: we first introduce the dataset employed in the experimentation and illustrate the annotation process devised to extend this data also for the sake of the identification task. We then describe the lexical resources employed in the experimentation; we discuss in detail how these were used in accord with their constructive rationale (that is, providing vectors for either senses or terms and senses). We then provide the obtained results, as regards as both the semantic similarity task and the sense identification task. We then discuss our results, and introduce further experiments to elaborate on technical details and features characterizing the employed embeddings. All experiments were run twice: by disregarding coverage issues (in this case each resource has been tested on the fraction of covered data), and by selecting the intersection of data covered

by all resources. We report and discuss all results, that allow for a more complete assessment of the results on such a wide and varied experimental base.

5.3.1 SemEval 2017 dataset

For the experimentation we chose the SemEval 2017 Task 2 – Subtask 1 English dataset (Camacho-Collados et al., 2017). The dataset is made of 500 word pairs (all of them nouns, that include named entities), originally annotated with a similarity score. In order to collect the 500 English pairs the authors chose 34 domains from the BabelNet semantic network: from each domain 12 words were sampled, requiring at least one multi-word expression and two named entities to be included. In order to pick up words possibly out of any pre-defined domain, the authors added 92 extra words, whose domain was not decided beforehand. Given the set of the initial 500 seed words, the pairs were generated so to ensure a uniform distribution of pairs across the similarity scale. The similarity scores are based on a five-point Likert scale —ranging from 0 which means "totally dissimilar and unrelated" to 4, which stands for "very similar"—, and each such score expresses the ground truth rating provided by human annotators.

5.3.2 Data Annotation with Sense Identifiers

In order to experiment on the sense identification task we annotated the terms in the SemEval-2017 Task 2 English dataset (Camacho-Collados et al., 2017) with sense identifiers from the BabelNet sense inventory. Three annotators fluent in English annotated the 500 word pairs; each record was originally composed by a triple $\langle t, u, y \rangle$, containing the term pair and a numeric score expressing the similarity between the considered terms. Each such tuple was extended with two sets, S_t and S_u, containing the BabelNet synsets for t and u that were found as the appropriate senses for the terms at stake, and compatible with the similarity score y.[6] However, in cases where multiple senses were reputed appropriate for either term, only the

[6] S is a set rather than a single sense, since multiple instances of overlapping senses can be detected in the BabelNet sense inventory, but each S only contains *equivalent* senses. On average over the 500 term pairs, each term was annotated with 1.087 senses. For example, the term radiator was annotated with two senses, bn:00065884n ('A mechanism consisting of a metal honeycomb through which hot fluids circulate') and bn:00065883n ('Heater consisting of a series of pipes for circulating steam or hot water to heat rooms or buildings').

Table 5.4: List of the sense-embeddings considered in the experimentation, along with abbreviation and the type of indexing adopted by each resource. The three rightmost columns report the parameters set that was employed in the experimentation.

Abbreviation	Full Name	Indexing Principle	α	β	γ
LLX [1]	LESSLEX	synset | term	0.4	0.8	0.9
N2V [2]	NASARI	synset | term	0.5	0.8	0.9
DCF [3]	DECONF	synset | term	0.5	0.6	0.9
SSE [4]	SENSEEMBED	⟨term_synset⟩ | term	0.5	0.3	0.8
SW2V [5]	SW2V	⟨term_synset⟩ | term	0.6	0.2	0.8
LSTMBD[6]	LSTMEMBED	⟨term_synset⟩ | synset | term	0.5	0.2	0.8

[1] (Colla, Mensa, & Radicioni, 2020a) (http://ls.di.unito.it v. 1.0)
[2] (Camacho-Collados et al., 2016) (http://lcl.uniroma1.it/nasari/)
[3] (Pilehvar & Collier, 2016) (https://pilehvar.github.io/deconf/)
[4] (Iacobacci et al., 2015) (http://lcl.uniroma1.it/sensembed/)
[5] (Mancini et al., 2017) (http://lcl.uniroma1.it/sw2v)
[6] (Iacobacci & Navigli, 2019) (https://github.com/iiacobac/LSTMEmbed)

most prominent sense was recorded by the annotator.

The three annotations were then merged through a simple voting strategy: we chose the senses selected by at least two annotators (*minimal consensus*). Alternatively, if no sense was found in BabelNet for either term, or no minimal consensus was reached on either term, the pair was dropped. Out of the 500 starting pairs we dropped 8 pairs, thereby resulting in a grand total of 492 annotated pairs. For the 984 terms therein, overall $15,558$ Babel synsets were found, corresponding to $144,262$ possible sense combinations, on average over 288 per each term pair. Such annotation obtained an averaged pairwise $.89$ Cohen's k inter annotator agreement on the individual terms, and $.79$ on term pairs. The dataset is described in (Colla, Mensa, & Radicioni, 2020c).[7]

5.3.3 Resources

We have then selected a set of recent and influential sense embeddings from literature, and used them for experimentation. They are listed in Table 5.4, along with the parameters employed in the experimentation. Parameters were optimized as follows. For each set of embeddings we recorded the accuracy obtained both in the semantic similarity task and in the sense individuation task. The experiments were run by varying the three considered parameters in the range $\{0, 0.1, 0.2, \ldots, 1\}$,

[7] The dataset is available on the Mendeley repository, https://data.mendeley.com/datasets/r5fbdpvnkk/1.

thereby resulting in 11 runs for each parameter, overall $1,331$ runs. Parameters were chosen so to ensure the best trade-off between word similarity and sense identification results. Such trade-off was computed as the balanced (F_1) harmonic mean of the $F_1(\rho, r)$ recorded in the word similarity task and the F_1(P,R) obtained in the sense identification task (please refer to the results reported in Table 5.5).

The resources used in the experimentation implement different sorts of indexing, either based on sense, or on ⟨term, sense⟩ pair, as it is shown in the third column ('Indexing Principle') of Table 5.4. In the following we briefly illustrate our strategy to handle all mentioned resources to compute the $\mathcal{R}$-sim and $\mathcal{N}$-sim metrics.

5.3.4 Sense Retrieval

Using the mentioned resources involves accessing embeddings therein in different ways. Specifically, given a term t we perform two steps: we first access BabelNet to retrieve all senses associated to t, thereby obtaining S^t, and we then retrieve the vector corresponding to each sense $s_i^t \in S^t$. While the first step is the same for all resources, in the second one different strategies were devised in order to cope with the type of indexing characterizing each resource. For those indexed on senses only, we directly retrieve the embedding corresponding to the sense s_i^t; alternatively, in resources providing an index for the pair ⟨term, sense⟩, we retrieve the embedding indexed through the pair $\langle t, s_i^t \rangle$. For example, given the term Wave and the word sense bn:00041739n which refers to "greeting", we retrieve the vector for bn:00041739n in resources indexed on senses only. On the other hand, for those indexed on ⟨term, sense⟩ pair, we retrieve the vector for Wave-bn:00041739n.

Based on the adopted type of indexing, the considered resources can be partitioned into two broad classes: NASARI2VEC, DECONF and LESSLEX provide sense descriptions, and they can be thus accessed by searching the input term in BabelNet, and through the retrieved sense identifier. Conversely, in SENSEEMBED, SW2V, and LSTMEMBED every sense representation is actually indexed on a pair ⟨term, sense⟩, so that different vectors correspond to a given sense identifier, one for each term. For example, the vector representing the term Wave-bn:00041739n differs from the vector representing Greeting-bn:00041739n. As regards as term-

sense indexed resources, when computing the ranked-similarity, and namely its sub-component dealing with the distance between a term t and its senses (Equation 5.4), we retrieve the sense identifiers from BabelNet, so to obtain the corresponding vector representations. However, in some cases the senses s_i^t returned by BabelNet have no corresponding vector associated to the term t in SENSEEMBED, SW2V, and LSTMEMBED. This fact may be detrimental to the coverage of such resources in this experiment. The main purpose of the experimentation is investigating how the proposed metrics compare to the max-similarity approach, rather than assessing the employed embeddings. This different setting would have implied penalizing uncovering models, e.g., by assigning mid-scale similarity values to pairs involving OOV terms, by basically injecting some noise for uncovered terms. We thus decided not to adopt this procedure, but rather to specify the coverage for each resource. Additionally, to the ends of comparing the proposed similarity metrics on exactly same input, we re-ran the experiments by only considering the fraction of data covered by all resources; of course this approach has the virtue of allowing a more precise comparison on same input, but it also suffers from a more limited experimental base. Such results are provided in Appendix A.2.

5.3.5 Results

We investigated how the proposed metrics compare with the maximization of the cosine similarity. The maximization approach is referred to as $\mathcal{M}$-sim in the following, and implements the standard maximization as described in Equation 4.1. Such results are compared against those obtained by employing $\mathcal{R}$-sim and $\mathcal{N}$-sim; the comparison drawn involves both tasks, semantic similarity and sense identification. The results on the SemEval-2017 English dataset are presented in Table 5.5.

$\mathcal{R}$-sim As regards as the correlation with human similarity ratings, we observe that in four out of six cases $\mathcal{R}$-sim obtained improved results with respect to the $\mathcal{M}$-sim metrics. The magnitude of such improvement is diverse, ranging from 1% (LSTMEMBED) to 14% (NASARI2VEC) in the (balanced, or F1) harmonic mean of Spearman's ρ and Pearson's r. In one case we recorded same correlation (DECONF), and in one case a slightly reduced (-2%, for SW2V vectors) correlation. As regards

Table 5.5: Results on the SemEval 17 English dataset. Reported figures express Pearson (r), Spearman (ρ) correlations and their F1 score, and Precision (P) and Recall (R) along with their F1 score for the proposed $\mathcal{R}$-sim and $\mathcal{N}$-sim metrics, that are compared to $\mathcal{M}$-sim. Numbers in brackets specify the coverage for each resource.

Resource [% cov]	Metrics	Semantic Similarity			Sense Identification		
		ρ	r	$F_1(\rho,r)$	P	R	F_1(P,R)
LLX [0.92]	$\mathcal{M}$-sim	0.73	0.73	0.73	0.33	0.31	0.32
	$\mathcal{R}$-sim	0.83	0.81	0.82	0.50	0.47	0.49
	$\mathcal{N}$-sim	0.83	0.80	0.81	0.47	0.84	0.60
N2V [0.70]	$\mathcal{M}$-sim	0.62	0.60	0.61	0.44	0.41	0.42
	$\mathcal{R}$-sim	0.75	0.75	0.75	0.60	0.56	0.58
	$\mathcal{N}$-sim	0.63	0.62	0.62	0.60	0.65	0.62
DCF [0.63]	$\mathcal{M}$-sim	0.80	0.79	0.79	0.64	0.60	0.62
	$\mathcal{R}$-sim	0.79	0.79	0.79	0.71	0.67	0.69
	$\mathcal{N}$-sim	0.81	0.80	0.80	0.69	0.84	0.76
SSE [0.61]	$\mathcal{M}$-sim	0.70	0.68	0.69	0.72	0.68	0.70
	$\mathcal{R}$-sim	0.74	0.71	0.72	0.83	0.78	0.80
	$\mathcal{N}$-sim	0.75	0.73	0.74	0.82	0.79	0.81
SW2V [0.86]	$\mathcal{M}$-sim	0.77	0.76	0.76	0.69	0.65	0.67
	$\mathcal{R}$-sim	0.75	0.74	0.74	0.76	0.71	0.73
	$\mathcal{N}$-sim	0.77	0.77	0.77	0.77	0.77	0.77
LSTMBD [0.69]	$\mathcal{M}$-sim	0.67	0.66	0.66	0.84	0.79	0.81
	$\mathcal{R}$-sim	0.68	0.67	0.67	0.88	0.83	0.85
	$\mathcal{N}$-sim	0.71	0.70	0.70	0.87	0.86	0.87

as sense identification, in all cases $\mathcal{R}$-sim allowed us to obtain improved precision and recall with respect to $\mathcal{M}$-sim; the magnitude ranges from 4% (LSTMEMBED) to 16.5% (LESSLEX).

Such figures suggest that not only $\mathcal{R}$-sim favorably compares to $\mathcal{M}$-sim in the semantic similarity task, but it produces a consistent improvement in the sense identification task. This has an important consequence since, as previously illustrated, the higher the performance in the sense individuation, the more reliable the results in the semantic similarity task.

$\mathcal{N}$-sim As regards as the semantic similarity task, the $\mathcal{N}$-sim metrics ensures a further gain over scores obtained through $\mathcal{R}$-sim in four out of six cases. The only relevant exception to this trend is the result of the experiment involving NASARI2VEC embeddings, in which we recorded a consistent drop, in the order of 13%. In the sense identification task, all resources obtained improved scores when employing the $\mathcal{N}$-sim metrics, from 1% in the case of SENSEEMBED, up to the 11.8% observed when experimenting with LESSLEX.

On average over all resources, the results obtained in experiments where $\mathcal{R}$-sim was employed show a 4.00% improvement over $\mathcal{M}$-sim in the semantic similarity task; likewise, results obtained by employing $\mathcal{N}$-sim show an average 3.43% improvement with respect to $\mathcal{M}$-sim. Also in this setting, both figures are supported by a more consistent advantage (10.29% and 14.57% for $\mathcal{R}$-sim and $\mathcal{N}$-sim, respectively over the $\mathcal{M}$-sim metrics) in the sense identification task, which guarantees more reliable correlations in the former task.

By and large, the proposed metrics improve on the maximization of cosine similarity; this holds for both considered tasks, although to a different extent across resources. Additionally, we can leverage different types of information available in the mentioned resources as detailed and discussed in the next Section.

5.3.6 Discussion

The figures obtained by experimenting on semantic similarity with $\mathcal{R}$-sim and $\mathcal{N}$-sim cannot be directly compared to state-of-the-art results on this dataset (0.79 F1 score) (Camacho-Collados et al., 2017; Speer & Lowry-Duda, 2017). To compare with SemEval results, we should have penalized uncovering models, e.g., by assigning mid-scale similarity values to pairs involving OOV terms. However, as mentioned, this would have determined injecting some noise for uncovered terms. Since our experimentation is aimed at investigating how the $\mathcal{R}$-sim and $\mathcal{N}$-sim metrics compare to $\mathcal{M}$-sim (rather than to assessing the sets of embeddings), we chose not to adopt this evaluation scheme: each result reported in Table 5.5 reflects the coverage of the considered dataset featuring a given set of embedding.

At a closer look, the constructive rationale underlying SENSEEMBED, SW2V and LSTMEMBED seems to be more precise: such resources obtained consistently higher scores in the sense identification task, showing that in this task the resources providing a representation for a pair ⟨term, sense⟩ overcome those relying on standard sense representation (that is, LESSLEX, NASARI2VEC, and DECONF).

This can stem from the fact that fewer senses are available —on average— in resources representing the pair ⟨term, sense⟩, which entails a reduced problem space. Some statistics describing the average number of senses per term, and the average size of neighborhoods featuring all employed resources are reported in Table 5.6.

Table 5.6: Statistics describing the number of senses available for each resource, along with the size of the neighborhood employed in the $\mathcal{N}$-sim metrics. The last two columns report the results in the semantic similarity and the sense identification tasks through the F_1 score of Spearman and Pearson correlation coefficients, and the F_1 score between precision and recall, respectively.

Resource	Measure	AVG term senses	AVG $\lvert\hat{S}\rvert$	Semantic Similarity	Sense Identification
				$F_1(\rho,r)$	F_1(P,R)
LLX	$\mathcal{M}$-sim	15.68	3.68	0.73	0.42
	$\mathcal{R}$-sim			0.82	0.58
	$\mathcal{N}$-sim			0.81	0.62
N2V	$\mathcal{M}$-sim	13.70	1.32	0.61	0.42
	$\mathcal{R}$-sim			0.75	0.58
	$\mathcal{N}$-sim			0.62	0.62
DCF	$\mathcal{M}$-sim	3.89	1.63	0.79	0.62
	$\mathcal{R}$-sim			0.79	0.69
	$\mathcal{N}$-sim			0.80	0.76
SSE	$\mathcal{M}$-sim	5.34	1.06	0.69	0.70
	$\mathcal{R}$-sim			0.72	0.80
	$\mathcal{N}$-sim			0.74	0.81
SW2V	$\mathcal{M}$-sim	5.02	1.13	0.76	0.67
	$\mathcal{R}$-sim			0.74	0.73
	$\mathcal{N}$-sim			0.77	0.77
LSTMBD	$\mathcal{M}$-sim	2.22	1.12	0.66	0.81
	$\mathcal{R}$-sim			0.67	0.85
	$\mathcal{N}$-sim			0.70	0.87

Conversely, the improvement obtained by employing $\mathcal{R}$-sim and $\mathcal{N}$-sim in the semantic similarity task is less consistent.

If we focus on results on the semantic similarity, we observe that the $\mathcal{R}$-sim provides the highest improvements —w.r.t. the $\mathcal{M}$-sim metrics— in resources adopting sense representation, while resources adopting the ⟨term, sense⟩ indexing obtain less consistent improvement. This fact may be due to the reduced number of senses per term, that is affected by the corpus employed to train such models: if the corpus only contains the most frequent senses for each term, then the model learns only the same most frequent ⟨term, sense⟩ pairs. Even though this phenomenon should be asymptotically absent as the size of training corpora will grow in future, to date this sort of reduced semantic coverage seems a plausible explanation for the limited impact of proposed metrics when using this class of resources in the semantic similarity task.

In order to further elaborate on the kind of effect and impact of the indexing principle, we re-ran the experiment involving the LSTMEMBED resource. In the first execution we employed the ⟨term, sense⟩ indexing, while in the second run the

Table 5.7: Comparison of the results obtained with the two different experimental settings adopted for LSTMBD: with the sense indexing only (LSTMBD$_S$) and with the previously adopted $\langle term, sense \rangle$ indexing (LSTMBD$_{S,T}$). Reported figures express Pearson (r), Spearman (ρ) correlations and their F1 score for the semantic similarity task, and Precision (P) and Recall (R) along with their F1 score for the sense individuation task. Statistics describing the number of senses available along with the size of the neighborhood employed by the $\mathcal{N}$-sim metrics.

Resource [% cov]	Measure	AVG term senses	AVG $\lvert\hat{S}\rvert$	Semantic Similarity	Sense Identification
				$F_1(\rho,r)$	F_1(P,R)
LSTMBD$_{T,S}$ [0.69]	$\mathcal{M}$-sim	2.22	1.12	0.66	0.81
	$\mathcal{R}$-sim			0.67	0.85
	$\mathcal{N}$-sim			0.70	0.87
LSTMBD$_S$ [0.82]	$\mathcal{M}$-sim	4.85	1.20	0.70	0.67
	$\mathcal{R}$-sim			0.73	0.79
	$\mathcal{N}$-sim			0.74	0.80

sense-level characterization alone was adopted. Results are reported in Table 5.7. Our findings show that employing the sense indexing only (bottom of Table) involves dealing with a higher number of senses per term: as expected, in such an experimental condition we obtained a relevant gain in the semantic similarity, at the expense of a less consistent improvement in the sense identification task. Also importantly, by adopting sense indexing, we obtained a considerable growth in terms of coverage (from 69% with $\langle$term, sense$\rangle$ indexing, to 82% when employing only senses), which may be a relevant *datum* for practical uses. Finally, a quick look at the results of the experiments performed by considering the fraction of data covered by all resources, reported in A.2 in Table A.7. These results are also complemented by the statistics on the average number of term senses and on the neighborhood size (Table A.8). The obtained results basically show similar trends, thus corroborating the previous findings. We summarize our results according to the task. As regards as the semantic similarity task, on average over all employed resources, $\mathcal{R}$-sim obtained a 2.57% improvement over $\mathcal{M}$-sim; the improvement of $\mathcal{N}$-sim over $\mathcal{M}$-sim is in the order of 1.29%. Interestingly, both figures are supported by a more consistent advancement (8.86% and 12.57% for $\mathcal{R}$-sim and $\mathcal{N}$-sim, respectively) in the sense identification task.

The ultimate synthesis of our experimentation is as follows. On average both $\mathcal{R}$-sim and $\mathcal{N}$-sim outperform $\mathcal{M}$-sim by a small margin in the semantic similarity task, while obtaining consistently more reliable scores in the sense identification task. Additionally, a supplemental experiment on LSTMEMBED has clearly

shown that a trade-off exists between results in the two tasks: resources indexed on ⟨term, sense⟩ basis ensure higher correlation with human judgment when identifying the senses, while resources indexed through senses only better correlate on semantic similarity. This trade-off also shows that obtaining even a small improvement in the semantic similarity while also always (in resource-independent fashion) identifying senses more reliably can be considered as a merit for $\mathcal{R}$-sim and $\mathcal{N}$-sim.

6 SE-MACAROON

SE-MACAROON (Sense Embeddings from MAny ContextuAl RepresentatiONs) is the second lexical resource that we developed. It consists of a set of distributional vectors that have been built by merging WordNet and BERT contextual word embeddings. Once again, we show how injecting a semantic layer on top of a language model can be beneficial in addressing tasks such as Word Sense Disambiguation.

The main contributions of this work are: (i) we devise a novel approach for constructing contextual sense embeddings, which characterizes each word sense through multiple vector descriptions; (ii) we build SE-MACAROON sense embeddings as an integration between WordNet and the BERT language model; (iii) we develop a novel Word Sense Disambiguation strategy.Moreover, we experimentally prove that representing word senses through multiple vectorial descriptions improves the accuracy in the Word Sense Disambiguation task over most competing approaches.

The chapter is organized as follows. In Section 6.1 we introduce the overall approach to build SE-MACAROON: we start by illustrating the procedure to retrieve the initial contexts (Section 6.1.1), then we detail the steps to build context sensitive word representations (Section 6.1.2). Finally, we introduce the contextualized sense embeddings generation (Section 6.1.3). Afterwards, we illustrate the experimental evaluation of SE-MACAROON vectors (Section 6.2): herein, we first present the statistics describing the resource (Section 6.2.1), and we then introduce the Word Sense Disambiguation task together with the evaluation benchmark employed (Section 6.2.2). We finally describe the whole WSD procedure devised (Section 6.2.3) along with the employed evaluation metrics (Section 6.2.4). In the last part of the chapter, we present the results on the WSD evaluation framework (Section 6.2.5), and we discuss the obtained results on the whole experimentation together with an in-depth analysis of the approach parameters (Section 6.2.6).

6.1 Building SE-MACAROON

The algorithm for the generation of SE-MACAROON is based on an intuitive idea: collecting the contextual representation of word senses in a sense tagged corpus, to build context sensitive conceptual representations. The idea underlying the building rationale is that we can better account for the precision of contextual representation computed by language models by maintaining their independence (their *contextual* profile, and thus their representational precision) rather than mixing them into a single fixed representation. Our intuition is that the embedding of a word sense s expressed with the word w it may be closer to the vector for the same word sense s lexicalized with w in a similar context rather than the average of all the lexicalization of s. In this respect, we started from BERT contextual embeddings and we built new sense embeddings relying and indexed on the WordNet sense inventory. We choose BERT as our starting point for several reasons: it is to date the most popular contextual language model; it poses limited requirements, with respect to newer language models, and allows dealing with higher amount of training instances in reasonable time without any loss of information (Sanh, Debut, Chaumond, & Wolf, 2019). Our approach can be divided into the following three steps:

- **Context retrieval**, to collect all the relevant sentences from a sense tagged corpus.
- **Word embedding**, to compute the vectorial representation for each term in the sentences, given the context retrieved in the previous step.
- **Sense embedding**, to collect all the vector representation for relevant words.

In Figure 6.1 we provide an example of the whole process starting from the term *affect* as expression of the word sense wn:00137313v — defined as *have an effect upon;* in WordNet—.

6.1.1 Context Retrieval

Each concept in SE-MACAROON is represented by a collection of vectors generated by processing sentences with the BERT language model. We start from a sense tagged corpus SC, where each instance is represented as a pair $\langle W_i, S_i \rangle$, where W_i

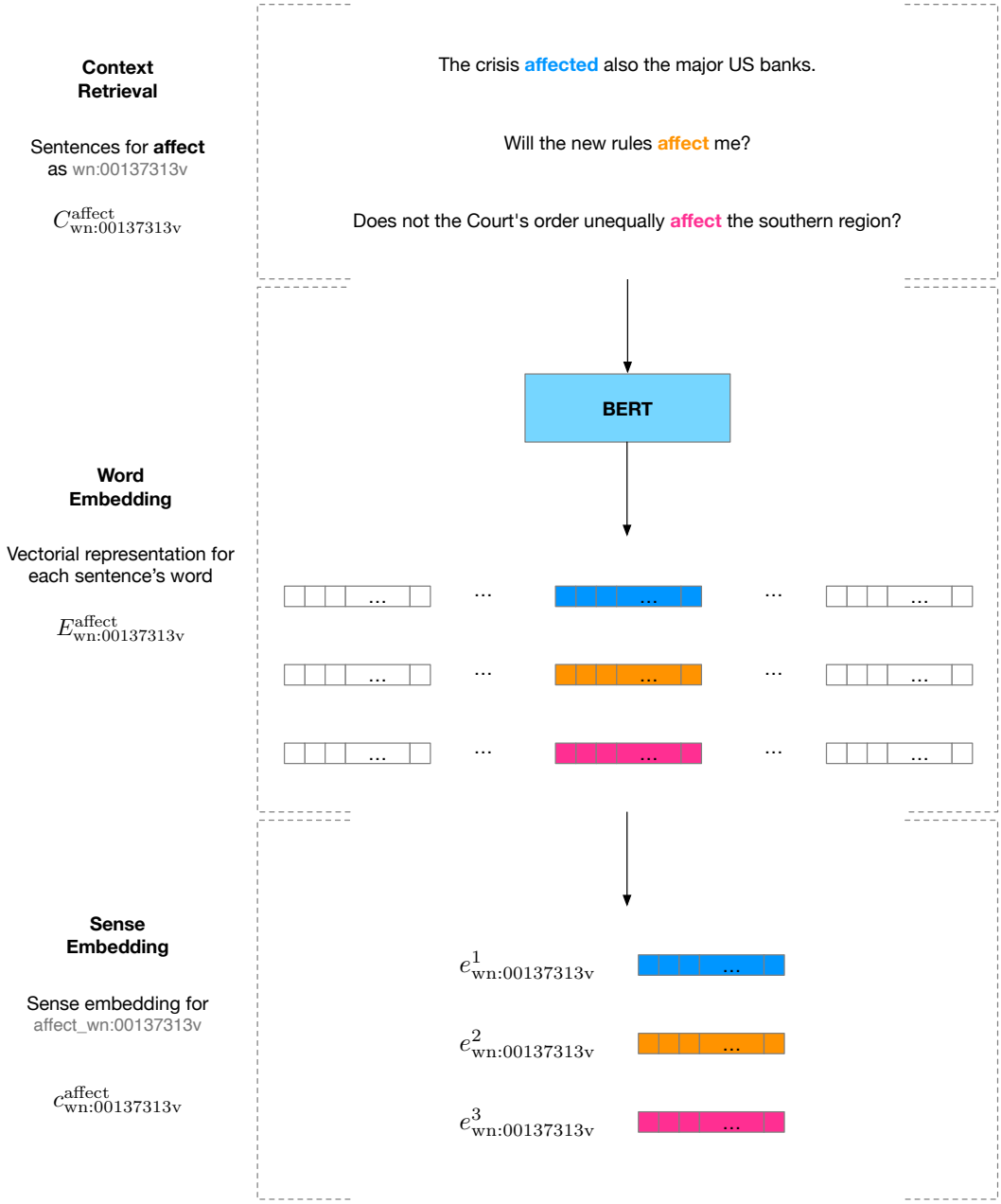

Figure 6.1: Graphical illustration of the working rationale of our approach. Let us consider the word *affect* as expression of the word sense wn:00137313v. We first retrieve all the sentences in which *affect* occurs as wn:00137313v, then process them with BERT and collect the contextual representation for *affect* in our SE-MACAROON.

is the sentence and S_i are the related senses. In particular, the words in the sentence $W_i = w_1^i w_2^i \dots w_k^i$ are associated with senses $S_i = s_1^i s_2^i \dots s_k^i$, where each s_j^i represents the sense with which the word w_j^i occurs in the sentence W_i. It is worth noting that, given the adopted sense inventory, not all words are actually equipped with a sense, that is, s_j^i may also be set to the null element. We adopted the WordNet3.0

sense inventory. For example, given the sentence:

The crisis affected also the major US banks.

its related senses in the WordNet sense inventory are:

$$\varnothing \ wn{:}13933560n \ wn{:}00137313v \ \varnothing \ \varnothing \ wn{:}01472628a \ wn{:}09044862n \ wn{:}08420278n$$

where each sense correspond to each word of the sentence, respectively. [1]

Given the word sense s and one of its lexicalizations w, we collect all the sentences, from a sense tagged corpus SC, in which w appears intended as s: in other words, we retrieve all the sentences where the sense s is lexicalized through the term w. More formally, given the pair $\langle w, s \rangle$, we define the set C_s^w containing all the pairs $\langle W_i, S_i \rangle$ from SC so that:

$$C_s^w = \bigcup_{\langle w,s \rangle} \{\langle w_j^i, s_j^i \rangle\} \qquad \text{with } w_j^i \in W_i \,|\, w_j^i = w \wedge s_j^i \in S_i \,|\, s_j^i = s.$$

In particular, considering the example reported in Figure 6.1, we define $C^{\text{affect}}_{\text{wn:00137313v}}$ as the set of all sentences from SC in which the term *affect* occurs with the sense wn:00137313v.

6.1.2 Word Embedding

The aim of the second step is to compute, by means of BERT, the representation of words in sentences of C_s^w. First, we pre-process sentences from C_s^w with the BERT tokenizer. Since BERT relies on the WordPiece tokenizer (Wu et al., 2016), sentence words might have been divided into sub-words, such as, for example, the word *unequally* is decomposed to the following tokens: ⟨ *une* ⟩, ⟨ *##qual* ⟩, ⟨ *##ly* ⟩, this means that both tokens *##qual* and *##ly* are a sub-words following their preceding token. Additionally, the BERT model is able to deal with sentences at most 128 tokens long. Given that, after tokenization, we filter C_s^w retaining only sentences with more than 10 words for which the last token corresponds to the end of the sentence.[2] Filtering sentences with less than 10 words allows us to remove short

[1] Senses are represented as WordNet synset offsets, while the null element is represented with the symbol ∅.

[2] That is, sentences are retained whose length is less than or equal to 128 tokens, which is actually the maximum size allowed by BERT tokenizer.

sentences for which contextual representations might be less precise than those for longer sentences. Additionally, since words might be divided in sub-tokens by the tokenizer, together with the maximum length for sentences, it may happen that some sentences exceed the tokens number limit, thus resulting in an inconsistent contextual representation. In this respect, it is necessary to check that the last token is consistent with the last word of the original sentence to ensure full detailed contextual representation for each such word. For example, given the sentence:

Does not the Court's order unequally affect the southern region?

the BERT tokenizer produces the following tokenization:

[⟨Does⟩, ⟨not⟩, ⟨they⟩, ⟨Court⟩, ⟨'⟩, ⟨s⟩, ⟨une⟩, ⟨##qual⟩, ⟨##ly⟩, ⟨affect⟩, ⟨the⟩, ⟨southern⟩, ⟨region⟩, ⟨?⟩]

Once sentences have been tokenized we then process the new representations with the BERT model, thus obtaining a contextual vectorial representation for each such token. The contextual embedding of an input word was computed as the average of its sub-token embeddings, that is, we define the vector for *unequally* as the average of the vector for *une*, *##qual* and of the vector for *##ly*. Since the BERT model is made of stacked layers, the typical word representation for semantic tasks is computed by summing the last four hidden layer embeddings (Jawahar, Sagot, & Seddah, 2019; Tenney, Das, & Pavlick, 2019; Tenney, Xia, et al., 2019). Given that, our word embeddings are computed as the sum of the last four hidden layer vectors. Finally, given the sentences in C_s^w, we define E_s^w as the contextual embedding for each word of each sentence in C_s^w, that is, E_s^w is defined as follows:

$$E_s^w = \text{BERT}(W_i) \;\; \forall W_i \in C_s^w \tag{6.1}$$

where $\text{BERT}(W_i)$ is the embedding function that produces contextual representation e_i^j for each word in the sentence. Therefore, for the sentence $W_i = w_1^i, \dots w_k^i$, the function $\text{BERT}(W_i)$ is defined as $\text{BERT}(W_i) = e_1^i e_2^i \dots e_k^i$ where e_i^j is the contextual representation for the j-th token in the i-th sentence W_i.

6.1.3 Sense Embedding

In this final step, we build a representation for each target sense in the sense tagged corpus; starting from the set of embeddings E_s^w we retrieve all the representations for the sense s lexicalized with the word w and collect them as our representation for the word sense. In particular, we define c_s^w as the collection of all the embeddings e_j^i from E_s^w that encode ($\leftarrow\!\!-$ in the Equation 6.2) the word w conveying the sense s:

$$c_s^w = \bigcup_{W_i,S_i} \{e_j^i \leftarrow\!\!- \langle w_j^i, s_j^i \rangle\} \qquad \text{with } w_j^i = w \wedge s_j^i = s;\ \ w_j^i \in W_i,\ s_j^i \in S_i. \tag{6.2}$$

Here w_j^i and s_j^i are the word and its related sense in the i-th dataset instance $\langle W_i, S_i \rangle$. Let us consider the example in Figure 6.1, the representation for $c^{\text{affect}}_{\text{wn:00137313v}}$ is defined as the collection of the three word embeddings for *affect* resulting from the application of the BERT model on the three sentences in which *affect* is intended as expression for the word sense wn:00137313v. At the end of the three steps each sense in the sense tagged corpus SC is provided with a set of associated vectorial representations, made of the collection of lexicalizations of the given sense.

It is worth noting that the sense embeddings c_s^w included in SE-MACAROON are indexed on both sense s and lexicalization w. That is, the sense embeddings c_s^w representing the exact expression of the word sense s lexicalized with the word w, are actually the collection of all the occurrences of s expressed by means of w, in contrast to the indexing principle adopted in the LESSLEX dictionary.

6.2 Evaluation

In this section we report the experimental settings in which we conducted the evaluation of SE-MACAROON when testing on English Word Sense Disambiguation task. In what follows we introduce the train set along with some resource's figures; then we introduce the test sets and the system setup. We then present the results along with the discussion.

Table 6.1: Figures on the generation process of SE-MACAROON, divided by Part of Speech. The average occurrences per sense are reported together with their standard deviation (σ).

SE-MACAROON Statistics	All	Nouns	Verbs	Adjs	Advs
Sense (term, synset) Vectors	31,352	14,893	8787	5939	1733
Senses Occurrences	199,363	76,062	78,820	27,787	16,694
AVG Occurrences per Sense	6.36 (±66.72)	5.11 (±50.48)	8.97 (±105.00)	4.68 (±14.79)	9.63 (±43.83)
Occurrences per Sense Range	[1, 9013]	[1, 5846]	[1, 9013]	[1, 413]	[1, 1475]
Sense (synset) Vectors	24,528	12,418	5794	5016	1320

6.2.1 SE-MACAROON Statistics

We trained SE-MACAROON sense embeddings on SemCor (G. A. Miller et al., 1994). SemCor is a manually sense-tagged corpus, divided in 352 documents for a total of 226,040 sense annotations and is, to our knowledge, the largest corpus manually annotated with WordNet senses. In this setting we adopted the BERT Large model since it is more accurate on modeling the English language with respect to BERT Base or their multilingual porting. We started from a total of 37,176 sentences contained in SemCor; we then retained 33,399 after the filtering step, thus resulting in 199,363 sense annotations. The final figures of the resource and details concerning its generation are reported in Table 6.1. The final number of sense embeddings in SE-MACAROON amounts to 31,352, corresponding to 24,548 unique synsets, covering only about the 21% of the total 117,659 WordNet synsets. The number of occurrences per sense ranges from 1 to 9013 for the verb *to be*.

6.2.2 Evaluation Benchmarks

We assessed SE-MACAROON vectors on the Word Sense Disambiguation task as it constitutes the most popular and obvious task for evaluating sense embeddings (Loureiro et al., 2022). WSD is a long-standing challange in the Natural Language Processing, as it lies at the core of language understanding (Navigli, 2009). WSD is defined as the task of associating words in context with its most suitable meaning from a pre-defined sense inventory. More formally, given the target word w_t and the sentence $W = w_1 \dots w_t \dots w_l$ the task consists in associating the sense underlying w_t in W. For example, given the word *bark* and the sentence *The tree's*

bark was dark. Let us consider WordNet as our reference sense inventory, in such example the task is to provide the correct entry form the WordNet's senses for *bark*, that is wn:13162297n defined as *tough protective covering of the woody stems and roots of trees and other woody plants*.

We carried out the evaluation on the test sets in the English WSD framework (Raganato, Camacho-Collados, & Navigli, 2017). The benchmark includes five standardized evaluation datasets from the past Senseval-SemEval competitions, that are: Senseval-2 (SE02) (Edmonds & Cotton, 2001), consisting of 2283 sense annotations, Senseval-3 (SE03) (Snyder & Palmer, 2004), consisting of 1850 sense annotations, SemEval-2007 (SE07) (Pradhan, Loper, Dligach, & Palmer, 2007), consisting of 455 sense annotations, SemEval-2013 (SE13) (Navigli, Jurgens, & Vannella, 2013), consisting of 1644 sense annotations, and SemEval-2015 (SE15) (Moro & Navigli, 2015), consisting of 1022 sense annotations. The benchmark also include the concatenation of the five test sets called ALL dataset.

6.2.3 System Setup

The SE-MACAROON word sense disambiguation strategy can be divided in the following three steps: *(i)* sentence embedding, aimed at computing contextual representations for the input sentence, *(ii)* sense occurrences scoring, aimed at computing a score for each sense's occurrence, and *(iii)* word sense ranking, where sense occurrences are ranked based on their score and the word sense is selected through a majority voting among the top N entries. The working rationale of the WSD strategy is presented in Figure 6.2.

Sentence Embedding The aim of the first step is to compute contextual representation for the input sentence. Therefore, given the sentence $W = w_0 w_1 \dots w_t \dots w_k$, containing the target word w_t, we process the tokenized sentence with BERT thus obtaining contextual embeddings $E = e_1 \dots e_t \dots e_k$ for each token in W. Once again, we average the sub-token vectors so to compute a token-level representation. Additionally, as for the building approach, we define the embedding e_i for a word w_i as the sum of the last four layers of the BERT model. Since we are interested in assessing the contribution of the other words in the sentence, particularly

those surrounding the target word w_t, we retain the contextual representation for those words occurring within a given span d from w_t only. That is, we define a context window surrounding the target word w_t, $CTX_{w_t} = e_{t-d}, \dots e_{t-1}, e_{t+1}, \dots e_{t+d}$ as the d word representations preceding and following w_t. Let us consider the Figure 6.2, the input sentence is *We must believe we have the ability to affect our own destinies: otherwise why try anything?* where *affect* is the target term. In the example our context window size is set to 3, which means that we retain the contextual embeddings for the three words preceding w_t and for the three words following w_t, we say that the left context is *the ability to* and our right context is *our own destiny*, and the $CTX_{\text{affect}} = [e_6, e_7, e_8, e_{10}, e_{11}, e_{12}]$.

Senses occurrences scoring After having built the contextual representation for the target word, along with its context, we compute a score for each occurrence $e_s^i \in c_s^{w_t}$ of word senses for our target word w_t to retrieve the most likely sense. In order to disambiguate w_t, we start by retrieving all the senses for the target word in the WordNet sense inventory, thus obtaining the set of candidate senses $S^{w_t} = s_1^{w_t}, s_2^{w_t}, \dots s_n^{w_t}$ as all the word senses for w_t for which a correspondence can be found in SE-MACAROON. Since we refer to senses for w_t, in what follows the superscript w_t is dropped to simplify the notation: s_i will thus be intended as $s_i^{w_t}$. For the sake of the readability, we recall here that each SE-MACAROON entry $c_{s_i}^{w_t}$ is defined as $c_{s_i}^{w_t} = [e_{s_i}^1, e_{s_i}^2, \dots e_{s_i}^n]$, that is, the set of occurrences of the word senses s_i expressed by the word w_t, thus we retain only WordNet's senses $s_i^{w_t}$ with a corresponding SE-MACAROON entry $c_{s_i}^{w_t}$. The scoring step is aimed at computing a score for each such sense occurrence $e_{s_i}^j$, expressing the semantic similarity between $e_{s_i}^j$ and w_t along with its related context CTX_{w_t}. Therefore, given a candidate word sense s_i and its occurrence $e_{s_i}^j$ for the target word w_t, we define the scoring function as follows:

$$\text{score}(e_{s_i}^j) = \frac{1}{2d+1}[\alpha * \text{sim}(e_t, e_{s_i}^j) + \sum_{k \in [t-d, \dots t-1, t+1, \dots t+d]} \frac{(1-\alpha)}{2d} * \text{sim}(e_k, e_{s_i}^j)], \quad (6.3)$$

where e_t and e_k indicate the BERT representation for the target word and the words in the context CTX_{w_t} respectively, while sim is the cosine similarity function. The α parameter is used to tune up the balance between the relevance of the similar-

ity among the word sense and the target word representation, and the similarity between the word sense and the context's words representations. We therefore compute a score for each occurrence $s^j_{s_i}$ of the candidate word sense s_i from S^{w_t}.

Let us consider the Figure 6.2, showing that SE-MACAROON contains three different sense representations for *affect*: $c^{\text{affect}}_{\text{wn:00137313v}}$, $c^{\text{affect}}_{\text{wn:00019448v}}$ and $c^{\text{affect}}_{\text{wn:00838043v}}$. Since each SE-MACAROON entry is provided with multiple occurrences, we compute the similarity between the contextual representation from e_6 to e_{12} [3] and each word sense occurrence $e^j_{s_i}$. We therefore compute a score for $e^j_{s_i}$ according to Equation 6.3: the similarity between e_9 and $e^j_{s_i}$ is weighted by α, while each similarity score defined between the context's words and $e^j_{s_i}$ is equally weighted by $\frac{(1-\alpha)}{6}$, where $d = 3$, thus obtaining $2d = 6$.

Word senses ranking Once we have computed a score for each occurrence for senses in S^{w_t} (employing the formula presented in Equation 6.3) we can proceed with the ranking step. We therefore define the ranking R as the list of occurrences of senses $e^j_{s_i}$ sorted in descending order, based on their associated score. In order to retrieve the most likely word sense for w_t we adopt a majority voting strategy on the top N items of the ranking R. More precisely, we define a window RW containing the top N ranked items as $RW = [e^j_{s_i}, \dots e^l_{s_k}]$, we then count the number of times in which a sense occurs in RW and select the top scoring sense. More formally, for each word sense s_i in S^{w_t}, we define the following voting function:

$$\text{vote}(s_i, R) = \sum_{e^j_{s_k} \in RW} \text{count}(s_i, e^j_{s_k}), \tag{6.4}$$

where the count function returns 1 if $s_i = s_k$, that is:

$$\text{count}(s_i, e^j_{s_k}) = \begin{cases} 1 & \text{if } s_i = s_k \\ 0 & \text{otherwise} \end{cases}. \tag{6.5}$$

Once all the senses from S^{w_t} have been provided with a score through the voting function, we select the most voted word sense for our target word w_t. More

[3] e_9 is the contextualized embedding for the target word *affect*, while $[e_6, e_7, e_8, e_{10}, e_{11}, e_{12}]$ represent the context CTX_{affect}.

formally, to retrieve the most voted word sense, we select s_* as:

$$s_* = \arg\max_{s_i \in S^{w_t}} \text{vote}(s_i, R), \tag{6.6}$$

that is, s_* is the most voted word sense in the top N elements of the ranking R. Let us consider again the example reported in Figure 6.2: here the ranking R is built by sorting the occurrences for the three senses wn:00137313v, wn:00838043v and wn:00019448v for *affect*. In the example, the size N of window RW is set to 3, thus obtaining two votes for the sense wn:00137313v and one vote for wn:00838043v. Therefore, we select the sense wn:00137313v for the target word *affect* in our input sentence.

6.2.4 Evaluation Metrics

Precision (P), Recall (R) and their harmonic mean (F1) metrics have been largely accepted from the literature to assess the performances of systems on the Word Sense Disambiguation task. The mentioned metrics have been adopted to overcome the Accuracy drawback: the accuracy focuses on the positive class, giving no intuition on the system's performances on the negative class (Cohn, 2003; Edmonds & Cotton, 2001). Following the literature, we redefine the precision as the fraction of correctly predicted senses within the set of instances for which the algorithm provided an answer, while the recall is defined as the the proportion of correctly predicted senses within the set of benchmark instances. More formally, the employed metrics are defined as follows:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{|D|} \qquad F1 = 2 * \frac{P * R}{P + R} \tag{6.7}$$

where TP (True Positives) are the correctly predicted instances; FP (False Positives) denote the instances for which the system provided a wrong prediction out of the $|D|$ tagged instances in the evaluation benchmark. It is worth noting that $TP + FP$ corresponds to the number of instances for which the assessed resource could actually provide a prediction, therefore we define the sum of true positives and false positives as *Coverage*. We report this figure, as well, to complement the results recorded through the recall metrics: in fact, the recall scores as an error both

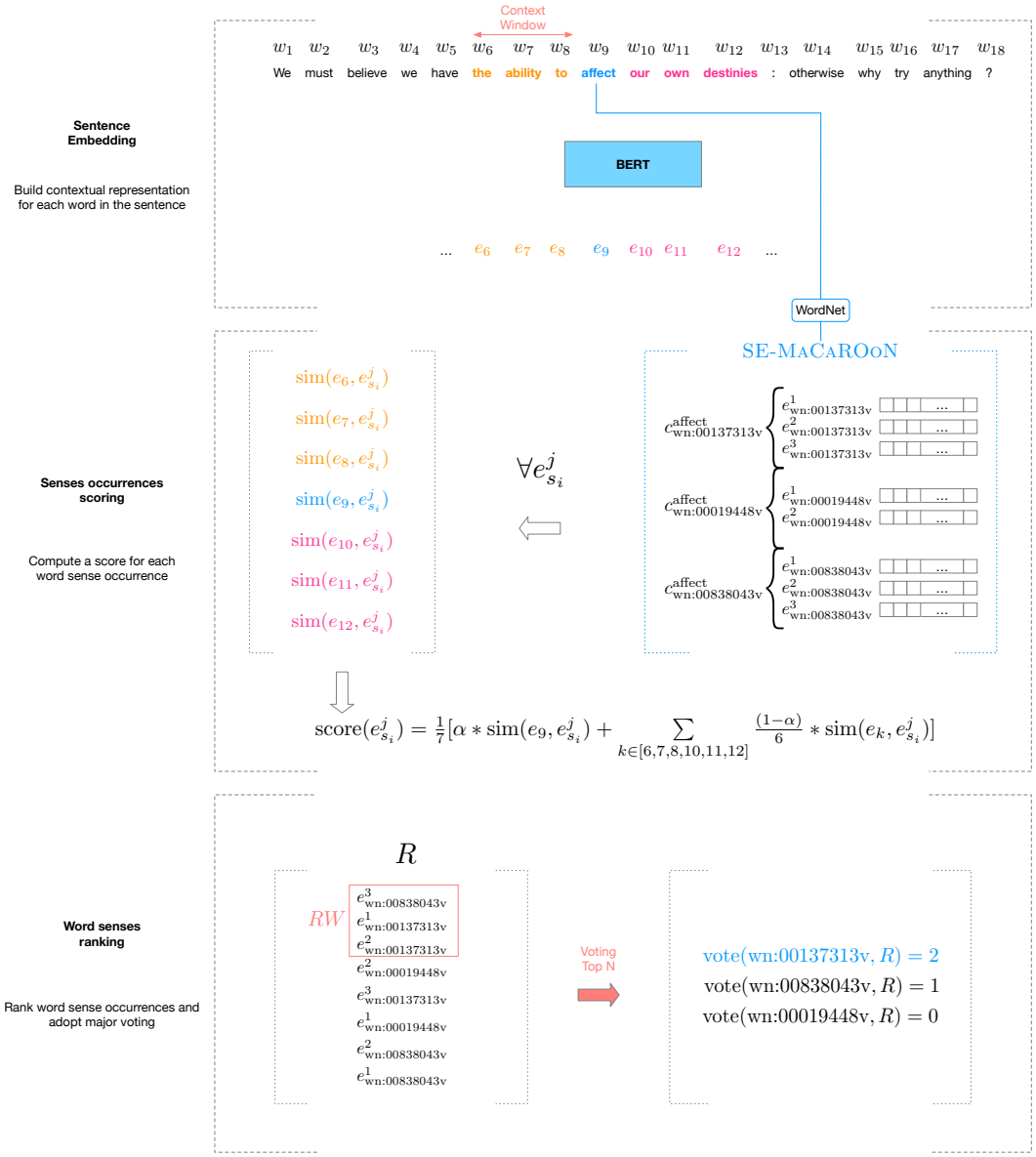

Figure 6.2: Graphical illustration of the working rationale of the WSD strategy. Let us consider the word *affect*, as our target word, occurring in the sentence *We must believe we have the ability to affect our own destinies: otherwise why try anything?*. At first we build contextual embeddings for each word in the sentence through BERT; then we retain only the representations for $W (W = 3)$ words from the left (left context, orange in the figure) and W words from the right (right context, pink in the figure) of the target word, along with the embedding for *affect*. We then compute the similarity between the context, including the target word, and every occurrence $e^j_{s_i}$ of each sense for *affect* in SE-MACAROON; we then define the score for e^i_s as the weighted average of the similarities. Eventually, we rank all the occurrences $e^j_{s_i}$ of each word sense for *affect* and extract, through majority voting, the sense wn:00137313v as the most likely sense considering the top N $(N = 3)$ occurrences of the ranking.

actual errors and uncovered senses for which the system was unable to provide a prediction.

6.2.5 Results

We compared SE-MACAROON with the most recent lexical resources representing word senses as contextualized embeddings. All the assessed resources built sense embeddings with the same language model, BERT Large.[4] Given the definition for the evaluation metrics provided in Equation 6.7 and according to the building principles underlying each resource, we adopted different disambiguation strategies. Since LMMS, SENSEMBERT, SENSEMBERT$_{sup}$, and ARES representations are twice as large as the BERT representations,[5] we repeated the BERT embedding (this step was implemented based on the literature presenting each and every employed resource) of the target word to match the number of dimensions. Conversely, LMMS-R embeddings match the number of dimensions of the representations produced through BERT.

For all the assessed resources, the adopted disambiguation strategy is the 1-nearest neighbor: for each target word w in the test set we computed its contextual embedding by means of BERT and compared it against the embeddings of the assessed resource associated with the senses of w. The Most Frequent Sense heuristics is customarily adopted in literature as the backoff strategy for non covered instances —i.e., predicting the most frequent sense of a lemma in WordNet for instances unseen at training time—, however, to assess the precision of each such resource we decided not to make use of the backoff strategy. On Table 6.3 we report the coverage for each resource and each benchmark in the evaluation framework.

Together with the state of the art resources, we also compared SE-MACAROON to SE-MACAROON$_{AVG}$ that has been devised by following the constructive rationale of our resource and by also averaging all sense occurrences into a single vectorial representation for each pair $\langle w_i, s_i \rangle$.

Parameters optimization Parameters employed by the SE-MACAROON and SE-MACAROON$_{AVG}$ systems are α (Equation 6.3); the size d of the context window for the target word CTX_{w_t}; and the size N of the ranking window RW, while the number of occurrences for each word sense was not limited. α was set to 0.5, d

[4] Available on the TensorFlow Hub repository at https://tfhub.dev/tensorflow/bert_en_cased_L-24_H-1024_A-16/4.

[5] BERT Large embeddings have 1024 dimensions while the resource's representations are provided with 2048 dimensions.

Table 6.2: Results on the SemEval 17 English dataset. Reported figures express Precision (P), Recall (R) metrics along with their harmonic mean F1 computed according Equation 6.7.

Resource	SE02 n = 2282			SE03 n = 1850			SE07 n = 445			SE13 n = 1644			SE15 n = 1022			ALL n = 7253		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LMMS	0.73	0.73	0.73	0.70	0.70	0.70	0.61	0.61	0.61	0.73	0.73	0.73	0.69	0.69	0.69	0.71	0.71	0.71
SENSEMBERT	0.80	0.44	0.57	0.76	0.43	0.55	0.75	0.30	0.43	0.73	0.73	0.73	0.75	0.46	0.57	0.76	0.50	0.60
SENSEMBERT$_{sup}$	0.83	0.46	0.59	0.83	0.47	0.60	0.75	0.30	0.43	0.77	0.77	0.77	0.76	0.47	0.58	0.80	0.52	0.63
LMMS-R	0.71	0.71	0.71	0.68	0.68	0.68	0.57	0.57	0.57	0.70	0.70	0.70	0.67	0.67	0.67	0.68	0.68	0.68
ARES	0.74	0.74	0.74	0.73	0.73	0.73	0.67	0.67	0.67	0.75	0.75	0.75	0.72	0.72	0.72	0.73	0.73	0.73
SE-MACAROON	0.76	0.68	0.72	0.74	0.69	0.72	0.67	0.59	0.63	0.77	0.67	0.71	0.71	0.61	0.66	0.75	0.67	0.70
SE-MACAROON$_{AVG}$	0.74	0.67	0.70	0.74	0.69	0.71	0.63	0.56	0.59	0.75	0.65	0.69	0.69	0.59	0.64	0.73	0.65	0.69

was set to 3 while N was set to 5.

The results obtained in the standard test sets of the WSD Evaluation Framework by Raganato et al. (2017) are reported in Table 6.2. We can see that the precision of SE-MACAROON is comparable with state of the art resources, such as, for example the precision of SE-MACAROON for SE02 is 0.76, SENSEMBERT$_{sup}$ obtained 0.83 at the cost of a lower recall 0.44, while ARES, which appears to be the top performing resource, obtained 0.74 as F1 score.

By examining the coverage reported in Table 6.3, we note that the coverage for both SENSEMBERT and SENSEMBERT$_{sup}$ is always close to the half of each benchmark, this is due to the fact that these resources have been built on nouns only. SE-MACAROON covers around 90% of each dataset, while both LMMS and LMMS-R together with ARES are able to deal with all instances in the evaluation framework. The low coverage of SENSEMBERT and partially for SE-MACAROON are reflected in lower recall scores: all the instances not covered are considered as same as errors when computing the recall. In fact, the recall for SE-MACAROON and SE-MACAROON$_{AVG}$ is lower when the coverage is under 90%, for example the recall for SE03 is 0.69 with a coverage of 93% while the recall for SE15 is 0.59 with a coverage of 0.86%. The F1 scores for SE-MACAROON are systematically higher than the F1 scores for SE-MACAROON$_{AVG}$ in a range from 1% to 4%.

6.2.6 Discussion

We overall experimented on five different datasets, included in the WSD Framework by Raganato et al. (2017). The obtained results authorise to state that SE-MACAROON is comparable with the state of the art, despite a slightly lower coverage. The improved performance of SE-MACAROON compared to SE-MACA-

Table 6.3: Coverage for each resource on the SemEval 17 English dataset. Reported figures express the absolute number of instances covered by the resource together with the percentage in square brackets [%].

Resource	SE02 n = 2282	SE03 n = 1850	SE07 n = 445	SE13 n = 1644	SE15 n = 1022	ALL n = 7253
LMMS [1]	2282 [1.0]	1850 [1.0]	455 [1.0]	1644 [1.0]	1022 [1.0]	7253 [1.0]
SENSEMBERT [2]	1262 [0.55]	1036 [0.56]	183 [0.4]	1644 [1.0]	633 [0.62]	4758 [0.66]
SENSEMBERT$_{sup}$ [2]	1262 [0.55]	1036 [0.56]	183 [0.4]	1644 [1.0]	633 [0.62]	4758 [0.66]
ARES [3]	2282 [1.0]	1850 [1.0]	455 [1.0]	1644 [1.0]	1022 [1.0]	7253 [1.0]
LMMS-R [4]	2282 [1.0]	1850 [1.0]	455 [1.0]	1644 [1.0]	1022 [1.0]	7253 [1.0]
SE-MACAROON	2052 [0.9]	1717 [0.93]	401 [0.88]	1432 [0.87]	874 [0.86]	6476 [0.89]
SE-MACAROON$_{AVG}$	2052 [0.9]	1717 [0.93]	401 [0.88]	1432 [0.87]	874 [0.86]	6476 [0.89]

[1] Loureiro and Jorge (2019a) (https://github.com/danlou/LMMS/tree/LMMS_ACL19)
[2] Scarlini et al. (2020a) (http://sensembert.org)
[3] Scarlini et al. (2020b) (http://sensembert.org)
[4] Loureiro et al. (2022) (https://github.com/danlou/LMMS)

ROON$_{AVG}$ seems to support the intuition that maintaining the independence of word senses occurrences may be beneficial in addressing the WSD task. The lower recall of SE-MACAROON with respect to full coverage resources such as LMMS, LMMS-R and ARES may be due to the number of unseen instances at training time. Despite that, we decided to test our hypothesis just relying on SemCor to exploit manual annotations, this allows us to analyze our results without dealing with silver or bronze data. In what follows we investigated the resource parameters together with the variables in our WSD approach. Namely, we assessed the impact of limiting the number of occurrences for each word sense, the effect of increasing the context window size d, the α balancing factor from Equation 6.3 and the impact of the size of the ranking window RW.

In the following experiments we adopted the precision as our evaluation metrics to get rid of instances for which SE-MACAROON is not able to provide a prediction, implicitly included in the F1 score through the recall metrics.

Number of senses occurrences In this respect, we first investigated the impact of the number of occurrences stored for each word sense, in particular we limited the number of vectors for each word sense in a range starting from 5 to 100.

The maximum number of occurrences per sense corresponds to $9,013$ vectors for the verb *to be* (please refer to Table 6.1), but on average the standard deviation is at most 105 for verbs: we thus decided to set the upper limit of our assessment to 100. The trend of the SE-MACAROON precision is depicted in Figure 6.3. In this setting, for those senses with less available occurrences than allowed by the limit,

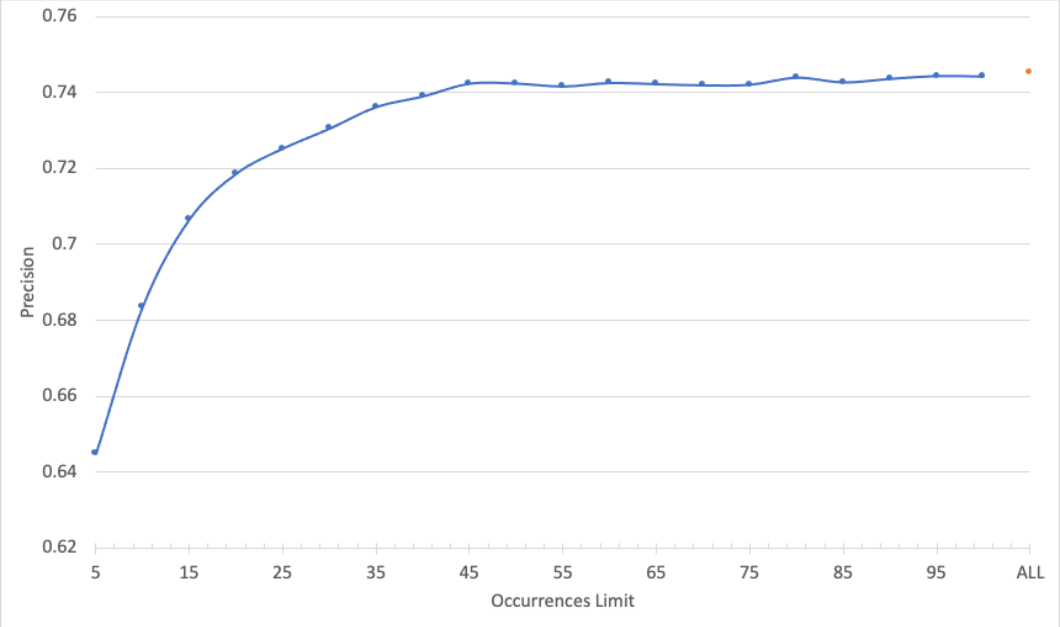

Figure 6.3: Precision of SE-MACAROON on the ALL concatenation when limiting the number of occurrences for each word sense. The precision starts from 0.64 when the limit is set to 5, and yields 0.74 when the limit is set to 100. The 0.75 presented in Table 6.2 is obtained by fixing no constraint to the number of occurrences for each word sense: this result is marked with the orange point. In this setting, we maintained fixed the dimension of the context window d to 3, the size of the ranking window RW to 5 and setting $\alpha = 0.5$.

we adopted the following aggregation strategy: given the word sense s lexicalized with w, and by fixing the limit to K occurrences for each word sense, we collected the first K representations form SemCor for the pair $\langle w, s \rangle$ in c_s^w; we then iteratively averaged the next occurrence of s with its nearest vector in c_s^w. According to this aggregation principle, the precision of SE-MACAROON systematically improves as the limit of occurrences increases: it starts from 0.64 when the limit is set to 5 to reach 0.74 when the limit is fixed to 100, while we obtain 0.75 when the number of occurrences is unbounded. This result, together with the improved F1 scores of SE-MACAROON with respect to SE-MACAROON$_{AVG}$, seems to support the intuition that storing different occurrences of the same word sense expressed with the same word may improve the precision of the resource, and bounding the number of vectors for each word sense to the standard deviation allows closely approaching the performance attained when setting no constraint. This bound enables a considerable reduction in the size of the resource.

Context Window Size Our assumption is that the context representation in which the target word w_t is placed may be helpful to disambiguate w_t. For the sake of readability, we recall here that we defined the context window $CTX_{w_t} = \{e_{t-d}, \ldots, e_{t-1}, e_{t+1}, \ldots e_{t+d}\}$ as the list of contextual embeddings for those words surrounding the target word w_t. In particular, saying that the context window size is d means to collect the representation for the d words preceding and following w_t. In Figure 6.4 we report the precision of SE-MACAROON when changing the size d of the context window, in a range from 0 to 10. It is worth noting that, when $d = 0$ we rely on the target word representation only, while when $d = 10$ we consider up to 10 words on both sides, hence at most 20 words in the context window. If the sentence contains less than d words preceding or following w_t, we still collect up to d available words on either side. We decided to adopt the range $[0, 10]$ for the context window size because of the average sentence length in SemCor: on average, the sentences in the sense tagged corpus contain $22(\pm 14)$ words, we therefore fixed the upper limit for d to 10 in order to fully account for the entire sentence representation on average. From Figure 6.4 we can see how the precision of the approach improves as we add words in context up to a maximum of 3, then starts a slow degrowth. Additionally, if we try to exploit the entire sentence representation — represented by the orange point— we obtain the same precision as using no context ($d = 0$). These results seem to support the intuition that although the representation of w_t provided by BERT is inherently contextual, we still enjoy a benefit considering also the local context, that is, taking into account the information provided by a few words surrounding the target w_t. It seems that by considering more than 3 neighbours preceding and following w_t may be detrimental in WSD: in most cases the word sense may be identified by taking into account few words rather than longer contexts that may be misleading.

α balancing factor The presented WSD approach strongly relies on the scoring function in Equation 6.3 which computes a score for each occurrence of word senses for the target word w_t. In particular, as the scoring function is computed balancing the similarity between the target word and the word sense occurrence $e^j_{s_k}$ and the similarity between the context and $e^j_{s_k}$ through the α parameter, we also investigated the effect of varying the balancing factor.

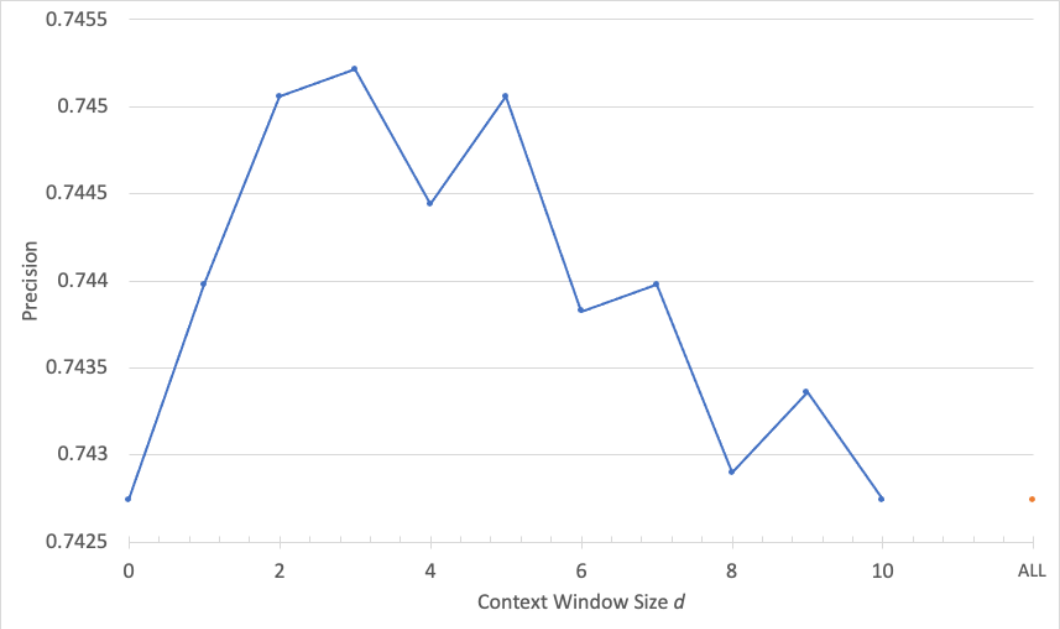

Figure 6.4: Precision of SE-MACAROON on the ALL concatenation when changing the size d of the context window. The orange point represents the precision of SE-MACAROON when considering the entire sentence as part of the context window. These results have been obtained by setting no limit to the number of occurrences for each sense, fixing the size of the ranking window RW to 5, $\alpha = 0.5$ and $d = 3$.

In Figure 6.5 we show the precision of SE-MACAROON when varying α in a range from 0 to 1 with 0.1 step. We note that $\alpha = 0$ means nullifying the left factor of the sum, systematically setting the similarity between the target word and the sense to 0 and accounting for the context similarity factor only. Conversely, when $\alpha = 1$ we are taking into account only the similarity between the target word w_t and the word sense occurrence. From the figure we can see how the precision of the approach improves as we increase α, obtaining the maximum with $\alpha = 0.5$ and starting to decrease with $\alpha = 0.6$. From this analysis it seems that the contribution of both similarity factors is relevant. Since the precision obtained when α ranges from 0 to 0.4 is lower than the precision when α is in the interval $[0.6, 1]$ we may assume that the contribution of the two factors is not equally distributed: the similarity between the target word and the word sense occurrence seems to be the most relevant, nevertheless the highest precision is obtained with $\alpha = 0.5$, this means that the contribution of the context similarity factor is not negligible.

Ranking Window size A relevant factor when disambiguating the target word with the proposed approach is the size of the ranking window RW in Equation 6.4.

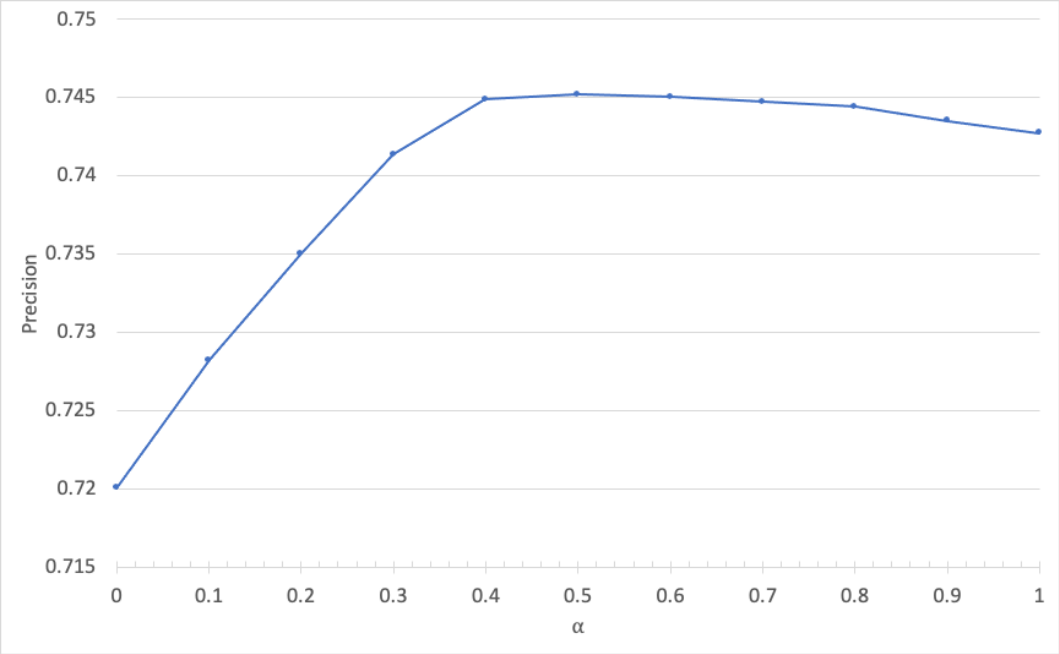

Figure 6.5: Precision of SE-MACAROON on the ALL concatenation when changing the α balancing factor from Equation 6.3. These results have been obtained by setting no limit to the number of occurrences for each sense, fixing the size of the ranking window RW to 5 and $d = 3$.

We recall here that in order to retrieve the most likely word sense for the target word w_t we compute the ranking R of all the word senses occurrences based on their score then we define a ranking window RW on the top N elements of R to restrict the candidates, and eventually we select the most likely word sense through majority voting. The size of the ranking window may affect the WSD performances, we then investigated the impact of varying the number of considered ranking items. In Figure 6.6 we report the precision of the WSD approach when varying the size of the ranking window, ranging from 1 to 10. We thus investigated the effect of limiting the assessment of the ranking window size to 10. From the figure we can see that when we consider the nearest neighbor ($|RW| = 1$) obtain a precision higher than 0.74, but considering two ranking's items the precision drops below 0.73, this may be due to the fact that, in case of tie on scores we chose the most frequent sense among the two most voted. When we consider more than two items the precision of the approach grows up to 5 occurrences obtaining a score close to 0.75, then starts to decrease until 10. These results seem to suggest that, on average, the occurrences of the correct word sense appears in the first five items of the ranking, thus supporting the hypothesis that the contribution of the different

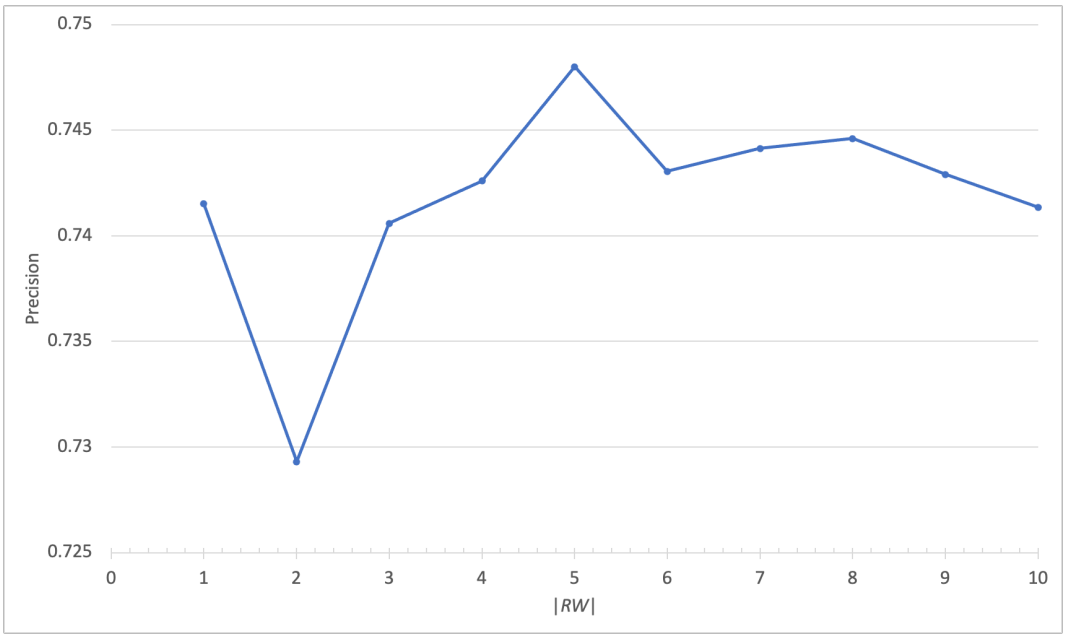

Figure 6.6: Precision of SE-MACAROON on the ALL concatenation when varying the size of the ranking window RW from Equation 6.4. These results have been obtained by setting no limit to the number of occurrences for each sense, fixing the size of the context window $d = 3$ and $\alpha = 0.5$.

word senses occurrences is not negligible.

7 Use Case: Using Language Models Perplexity as a Tool for Linguistic Analysis

After having illustrated the static embeddings of LESSLEX, the sense identification task, and the contextual —and still sense-oriented— SE-MACAROON embeddings, we now introduce an investigation in which language models have been employed to analyze language. This line of research relies on the *perplexity* metrics. Perplexity has been originally conceived as a tool for the intrinsic assessment of language models with respect to a sample of language. We now show how it can be employed for linguistic analysis purposes, such as to grasp differences stemming from linguistic registers and to categorize language from healthy vs. cognitively impaired subjects.

This chapter is structured as follows: in Section 7.2 we introduce related work and review the main approaches to automatically recognize subjects affected by different forms of psychotic disorders based on linguistic analysis. In Section 7.3 we provide the essential background to the perplexity metrics. We then describe the experiments devised to explore whether perplexity is a stable metrics, and whether it can be reliably used to detect mental disturbances (Section 7.4). To these ends, we first examine whether perplexity can be deemed as reliable to analyze speech transcripts under an intra-subject and discourse-level coherence perspective (Section 7.4.2); we then assess it by examining different subjects, and compare perplexity scores as computed through LMs built by employing different architectures (Section 7.4.3). Finally, we test perplexity to discriminate healthy subjects from subjects affected from Alzheimer Disease (Section 7.4.4). In the final section, we elaborate on the obtained results and illustrate future work to improve the perplexity-

based approach and make it a tool practically useful for diagnostic purposes.

7.1 Introduction

In economically developed societies the burden of mental disturbances is becoming more evident, with negative impact on people's daily life and huge cost for health systems. Whereas for many psychotic disorders no cures have been found yet, the treatment of people at high risk for developing schizophrenia or related psychotic disorders is acknowledged to benefit from early detection and intervention (Marshall et al., 2005). To this end, a central role might be played by approaches aimed at analyzing thought and communication patterns in order to identify early symptoms of mental disorder (Larson, Walker, & Compton, 2010).

The analysis of human language has recently emerged as a research field that may be helpful to analyze for diagnosing and treating mental illnesses. In fact, in the last decade Natural Language Processing (NLP) techniques have become a common tool to support research on psychotic disorders. Namely, if language and its associated cognitive functions are first impaired before the full signs of mental disorders become apparent, linguistic analysis assisted by computing systems may be helpful for early detection.

Recent advances in NLP technologies allow accurate language models (LMs) to be developed. In order to measure the distance between an actual sequence of tokens and the probability distribution we propose using *perplexity*, a metrics that is well-known in literature for the intrinsic evaluation of LMs. In this chapter we run experiments targeted at investigating how reliable perplexity is as a tool for investigating individuals' language, and we test whether the perplexity computed employing a language model acquired based on speeches from healthy subjects can be useful in discriminating healthy subjects from people suffering from mental disorders.

Although in literature perplexity is not new as a tool to compare the language of healthy and diagnosed subjects, this work is, to the best of our knowledge, the first attempt at analyzing how suited perplexity is to analyze individuals' spoken language. While the reliability of perplexity has been simply taken for granted in previous reports, we investigate whether and to what extent perplexity scores

are reliable before trying to use them to categorize stimuli. Moreover, as far as we know, no previous work has provided a comparison between perplexity scores computed through LMs as diverse as GPT-2 and Bigrams. This difference has practical consequences for applications, mostly due to the different computational effort required both to train and employ such models, and to the expressivity (and thus the descriptive power) of the learned models.

7.2 Related Work

Patients with psychiatric disorders such as schizophrenia show various semantic disturbances, and may suffer from difficulties in handling linguistic meanings at different processing levels such as morphology, syntax, semantics, and pragmatics (de Boer, Brederoo, Voppel, & Sommer, 2020). The work in (Covington et al., 2005) provides a rich overview on disturbances at the different levels. As far as we are concerned, disturbances related to schizophrenia typically produce abnormal usage of neologisms and word approximations, disruptions in language cohesion (Docherty, DeRosa, & Andreasen, 1996), syntactically simpler constructions featured by reduced use of embedded clauses and grammatical dependents (Çokal et al., 2018), inflectional morphology variants and errors (Walenski, Weickert, Maloof, & Ullman, 2010).

In the last decade, advances in NLP techniques have allowed the construction of automated approaches to computationally characterizing and predicting human behavior, including also many of the aforementioned linguistic levels. These approaches have identified markers that can help differentiate patients with psychiatric disorders from healthy controls, and predict the onset of psychiatric disturbances in high risk groups at the level of the individual patient.

Early work in this area started with generating vectors from co-occurrence matrices (Harman, 1993; Schütze & Pedersen, 1997), treated with latent semantic indexing (Landauer et al., 1998), or point-wise mutual information (Hindle, 1990). Such early distributional representations provided *explicit* (that is, directly meaningful and human-interpretable) information. The number of dimensions of such vectors was determined by the size of the vocabulary. On the other side, in *implicit* or latent representations, features were used resulting from Latent Semantic

Analysis (LSA). LSA is a multidimensional associative model based on the distributional hypothesis: word meaning is encoded as a multi-dimensional (usually 300 or 400 dimensions) vector obtained by elaborating large *corpora* to estimate the co-occurrence frequencies for each word. A basic approach based on LSA, such as that described in the seminal work by (Elvevåg, Foltz, Weinberger, & Goldberg, 2007), is as follows. Each input token is represented through a corresponding LSA vector, $W_i = \{I_{i1}, I_{i2}, \dots I_{iN}\}$. In turn, the vector representation for a phrase P is then built as the mean of the vectors representing all words in P: $P_i = \frac{1}{N}\sum_{k=1}^{N} I_{ik}$. The coherence between any two phrases is then computed through the cosine similarity of their corresponding vectors. The assumption underlying this approach is that meaningful texts will be featured by high coherence scores (in that words in the text being considered are semantically related on a distributional perspective), whilst text with some sort of disorder (or 'loose associations' among words) will be featured by reduced coherence scores. In (Bedi et al., 2015) an artificial dataset built by intentionally manipulating existing texts was used to test the described notion of coherence: the minimum semantic distance and the mean semantic distance of adjacent sentences were found to be negatively correlated with the disorder level introduced in the original. In this work LSA (in conjunction with information on grammatical Part-of-Speech function, referred to as POS tags) has been used to predict the transition to psychosis in a clinical high-risk cohort.

More recently, LSA techniques have been superseded by neural approaches aimed at learning latent representations of words called word embeddings (Navigli & Martelli, 2019). The overall design aimed at characterizing coherence (or, equivalently, the disorder associated with sentences and documents), by comparing vector representations of text excerpts, has remained unchanged.

A different approach to provide quantitative measures to language coherence and complexity is graph-based: in this setting, nodes represent words, and the word sequence is induced by directed edges. One main assumption underlying these approaches is that in coherent discourse neighboring words refer to connected topics, whilst incoherent discourse is associated with difficulties in making an ordered trajectory or path between topics. By employing tools from graph theory and information science it is possible to extract information on graph prop-

erties, such as connectedness, subgraphs or graph components. More specifically, measures such as entropy can be employed to probabilistically define topics and topic transitions (Cabana, Valle-Lisboa, Elvevåg, & Mizraji, 2011). Such graph representations also allowed grasping specific features of the normal and dysfunctional flow of thought (such as divergence and recurrence), and to produce accurate sorting of individuals affected by schizophrenia or mania (Mota et al., 2012). In another study, techniques for speech graph analysis were employed to describe formal thought disorder, which has been mathematically defined by the linear combination of connectedness graph attributes and their degree of similarity to randomly generated graphs. Such connectedness attributes were mapped onto a Disorganization Index, and used to classify negative symptom severity (Mota, Copelli, & Ribeiro, 2017).

In what follows we survey a set of works employing 'perplexity' that are specifically relevant to introduce our own proposal. Although originally conceived to assess how language models are able to model previously unseen data, perplexity can be used to compare (and discriminate) text sequences produced by healthy subjects or by people suffering from language-related disturbances. To provide a hint of this approach, perplexity is a positive number that —given a language model and a word sequence— expresses how unlikely it is for the model to generate that given sequence. A richer description of the perplexity is provided in Section 2.1.

In (Stolcke & Shriberg, 1996) N-grams of part of speech (POS) tags were employed to identify patterns at the syntactic level. Then, two LMs were acquired (one from patients' data and the other from data from healthy controls): the categorization of a new, unseen (that is, not belonging to either set of training data) sample was then performed through the perplexity computed with the two LMs over the sample. The considered sample was then categorized as produced by a healthy subject (patient) if the LM acquired from healthy subjects (patients) data attained smaller perplexity than the other language model. Perplexity has been recently proposed as an indicator of cognitive deterioration (Frankenberg et al., 2019); more specifically, the content complexity in spoken language has been recorded in physiological aging and at the onset of Alzheimer's disease (AD) and mild cognitive impairment (MCI) on the basis of interview transcripts. LMs used in this

research were built by exploiting 1-grams and 2-grams information; as illustrated in next section (please refer to Equation 2.2), such models differ in the amount of surrounding information employed. Perplexity scores were computed on ten-fold-cross-validation basis, whereby participants' transcripts were partitioned into ten parts; a model was then built by using nine parts and was tested on the tenth. This procedure was repeated ten times so that each portion of text was used exactly once as the test set. Four examination waves with an observation interval of more than 20 years were performed, and correlations of the perplexity score of transcriptions dating to the beginning of the experiment were found with the score from the dementia screening instrument in participants that lately developed MCI/AD.

In (Fritsch et al., 2019), perplexity has been employed as a predictor for Alzheimer Disease (AD) on the analysis of transcriptions from DementiaBank's Pitt Corpus (Becker, Boiler, Lopez, Saxton, & McGonigle, 1994), that contains data from both healthy controls and AD patients. More precisely, two neural language models, based on LSTM models, were acquired, one built on the healthy controls and the other trained on patients belonging to the dementia group. A leave-one-speaker-out cross-validation was devised and, according to this setting, a language model $\mathcal{M}_{-s}$ was created for each speaker s by using all transcripts from the speaker's group but those of s. Data from speaker s was then tested on both $\mathcal{M}_{-s}$, thus providing a perplexity score p_{own}, and on the language model built upon the transcripts from the whole group to which the speaker did not belong to, thus obtaining the perplexity score p_{other}. The difference between the perplexity scores $\Delta_s = p_{own} - p_{other}$ was computed as a description for the speaker s. The classification of each speaker was then performed by setting a threshold ensuring that both groups obtained equal error rate. The authors achieved 85.6% accuracy on 499 transcriptions, and showed that perplexity can also be exploited to predict a patient's Mini-Mental State Examination (MMSE) scores. The approach adopted in this work is the closest to our own work we could find in literature; however it also differs from ours in some aspects. First, we investigated how reliable perplexity is in assessing the language of healthy subjects. That is, we analyzed how perplexity scores vary within the same individual, as an initial step toward assessing if perplexity is suitable for examining text excerpts/transcripts that (like in the case of Pitt Corpus) were collected

through multiple interviews and tests, spanning over years. Additionally, we were concerned with evaluating all excerpts from a single individual to predict the AD diagnosis at the subject level, rather than in predicting the class for each and every transcript. In order to assess the perplexity as a tool to support the diagnosis, we analyzed only data from subjects for which at least two transcripts were available.

7.3 Perplexity

As mentioned, LMs are basically probability distributions of word sequences: perplexity was originally conceived as an intrinsic evaluation tool for LMs, in that it measures how far a model predicts a given word sequence (Goldberg, 2017). This measure is defined as follows. Let us consider a word sequence of k elements, $W = \{w_1, \ldots, w_k\}$; since we are interested in evaluating the model on unseen data, the test sequence W must be new, and not be part of the training set. Given the language model LM, we can compute the probability of the sentence W, that is $\mathrm{LM}(W)$. Such a probability would be a natural measure of the quality of the language model itself: the higher the probability, the better the model. The average log probability computed based on the model is defined as

$$\frac{1}{k}\log_2 \prod_{i=1}^{k} \mathrm{LM}(W) = \frac{1}{k}\sum_{i=1}^{k} \log_2 \mathrm{LM}(W),$$

which amounts to the log probability of the whole test sequence W, divided by the number of tokens in sequence. The perplexity is defined as

$$2^{-l}, \text{ where } l = \frac{1}{k}\sum_{i=1}^{k} \log_2 \mathrm{LM}(W);$$

that is, the perplexity of sequence W given the language model LM is 2 raised to the negative of the average log probability:

$$\mathrm{PPL(LM,W)} = 2^{-\frac{1}{k}\sum_{i=1}^{k} \log_2 \mathrm{LM}(w_i|w_{1:i-1})}. \tag{7.1}$$

It is now clear why low PPL values (corresponding to high probability values) indicate that the word sequence fits well to the model, or equivalently, that the model

is able to predict that sequence.

7.4 Experiments

After having introduced the notion of perplexity and a brief description on modern neural architectures, we explore whether —and to what extent— the perplexity of LMs attained through such architectures can be used as a biomarker to detect language anomalies. Language anomalies detection may be helpful in recognizing mental disturbances and other disorders. To these ends we need to investigate whether perplexity provides *reliable* (or *stable*) scores. Informally stated, by reliable we intend that similar text documents —such as repeated interviews to the same subject over a limited time span, or descriptions by different subjects about the same scene— should be featured by analogous perplexity scores (by employing the same language model). Reliability is defined based on two measurements: the absolute magnitude of perplexity scores, and the dispersion of such scores, measured through their standard deviation.

We are interested in exploring two focal questions: 1) whether perplexity scores are reliable and stable within the same subject, but still sensitive enough to account for different sorts of speech forms produced by a given speaker. Additionally, we test the discriminative power of perplexity, that is 2) whether the language of a specific class of subjects, diagnosed as suffering from disorders impacting on common linguistic abilities, can be automatically distinguished from that of healthy controls solely based on perplexity accounts.

In the first experiment, we analyzed whether the LMs acquired by re-training both Bigrams and GPT-2 on transcriptions of two different kinds of speech (two classes: political rallies vs. interviews) from a single subject produce different perplexity scores when the LM is used for analyzing similar (taken from same class) and different (from the other class) documents. In the second experiment we have measured the perplexity scores featuring discourses by 8 well-known political figures: in this case our aim was to assess whether such perplexity scores (computed either based on a general English LM or on a LM trained/fine-tuned on speeches from that subject) are stable within subject and across subjects. Finally, for the third experiment we have used the Pitt Corpus, from which we selected the transcripts

of responses to the Cookie Theft stimulus picture (Goodglass & Kaplan, 1983), and investigated whether the perplexity score allows discriminating patients with dementia diagnosis (n = 194) from healthy controls (n = 99).

7.4.1 Compared LMs

Three different experimental setups have been designed in order to compare perplexity as computed by language models acquired by training with two different architectures: Bigrams, and GPT-2.

Bigrams

Since Bigrams implement the simplest language model with context, where each word is conditioned on the preceding token only, we adopted the Bigrams model for the first experimental setup. The motivations underlying the decision to consider a single word only as context are both computational (i.e., by increasing the context size implies an increase in the training time too), and theoretical (that is, considering longer history involves dealing with higher data sparsity). In this setting, by following well-established approaches in literature (Jurafsky & Martin, 2014, Chap. 3), we define the probability of a sequence of words $W_{1,n} = w_1, w_2, \ldots, w_n$ as:

$$P(W_{1,n}) = \prod_{i=1}^{n} P(w_i|w_{i-1})$$

where the probability of each Bigram is estimated by exploiting the Maximum Likelihood Estimation (MLE). According to the MLE, we can estimate probability of the Bigram (w_{i-1}, w_i) as:

$$P(w_i|w_{i-1}) = \frac{C(w_i|w_{i-1})}{C(w_{i-1})} \tag{7.2}$$

where $C(w_i|w_{i-1})$ is the number of occurrences of the Bigram (w_{i-1}, w_i) in the training set, while $C(w_{i-1})$ counts the occurrences of the word w_{i-1} only. It is worth mentioning that training Bigrams on a limited vocabulary may lead to cases of out-of-vocabulary words, i.e., unseen words during the training process. Out-of-vocabulary words pose a problem in calculating the probability of the sentence in which they are involved: in such cases we are not able to compute the probability of the Bigram involving the unknown word, thus undermining the probability of the

whole sequence. In order to deal with out-of-vocabulary words, we replace each token occurring only once in the training set with the 'unknown' tag, UNK. In so doing, during the test phase we are allowed mapping each out-of-vocabulary word to the unknown word tag. Despite the strategy for handling out-of-vocabulary words, we may still end up with unseen Bigrams, formally occurring zero times in the training set, thus resulting in a null probability. We addressed the unseen Bigrams issue through the Laplace Smoothing technique (Jurafsky & Martin, 2014, Sec. 3.5.1), that is, adding one to all counts. According to the Laplace smoothing technique, we updated the Equation 7.2 as follows:

$$P(w_i|w_{i-1}) = \frac{C(w_i|w_{i-1}) + 1}{C(w_{i-1}) + V}$$

where V is the size of the vocabulary. In this setting, Bigrams have been computed through the Natural Language ToolKit (NLTK) package,[1] while the perplexity of a text has been computed according to Equation 7.1.

GPT-2

The second experimental setup that we designed exploits the GPT-2 neural model, in particular we used the GPT-2 pre-trained model available via the Hugging Face Transformers library.[2] In this setting, the input text has been preprocessed by the pre-trained tokenizer and grouped into blocks of 1024 tokens. The pre-trained model is specialized as Causal Language Model (CLM) on the input texts, that is, predicting a word given its left context. Since the average log-likelihood for each token is returned as the loss of the model, the perplexity of a text is computed according to Equation 7.1.

7.4.2 Experiment 1: Intra-subject and discourse-level coherence

The first experiment is aimed at investigating whether perplexity scores computed based on a given LM are stable, and whether perplexity scores are able to grasp factors specific to a given sort of speech. We have then targeted transcripts of two

[1] To compute Bigrams we exploited the *util* python package from NLTK http://www.nltk.org/api/nltk.html?highlight=ngrams#nltk.util.ngrams

[2] https://huggingface.co/gpt2

different kinds of discourse: the interview and the political rally. While in the former case both the questions put to the interviewee and his answers should convey a sense of poise, balance, and posture, political rallies are events where people sharing similar political beliefs gather to support their candidate. The language adopted in interviews is thus supposed to be more regular and consistent, whilst it should be more emphatic, direct and vehement in rallies. Our second research question was then whether the employed language models were able to recognize the two different linguistic registers.

Materials

We selected 10 transcripts by the former US President Donald Trump (this choice is mostly due to the large availability of his transcripts): 5 interviews and 5 campaign rallies were downloaded from the Rev platform.[3] Interviews were recorded between June 2019 and November 2020, while campaign rallies date to September and October 2020. The duration of both interviews and rallies varies between 45 minutes and one hour and 43 minutes. The statistics describing all transcripts employed in the first experimental setting are reported in Table A.9. While the initial choice of the transcripts was random within each category, we tried to select text excerpts of similar duration. The complete list and URLs of the employed material are provided in A.3.

Procedure

Two types of model were acquired, one for Political Rallies and one for Interviews, and this schema was replicated for both Bigrams and GPT-2.

Each LM was then tested on leave-one-out basis on transcripts in the same category as the training/fine-tuning, and in direct fashion on transcripts from the other category. In the following we will simply refer to training, even though in a strict sense training procedures were employed to acquire Bigram models, while fine-tuning[4] is associated to the refinement step of the base GPT-2 model. For example,

[3] https://www.rev.com.

[4] Our distinction seems compatible with a definition provided in literature: "In fine-tuning, we begin with off-the-shelf embeddings like word2vec, and continue training them on the small target corpus" (Jurafsky & Martin, 2014, p.399).

Table 7.1: Perplexity (PPL) scores along with standard deviations obtained with fine-tuning on the transcripts from the Rally and Interview categories, and averaged values for PPL scores and standard deviations.

Fine-tuning	Test	Bigrams		GPT-2	
		avg-PPL	avg-stdev	avg-PPL	avg-stdev
Rally	Interview	593.33	28.31	22.78	2.28
	Rally	582.02	24.70	19.44	0.69
Interview	Interview	430.88	27.39	22.20	1.30
	Rally	512.77	15.12	24.43	1.59

in order to compute the perplexity score for excerpts from the Rally category with a language model obtained by training/fine-tuning on the same category, 5 models were built by using 4 of the 5 available transcripts (the fifth one was used for testing); results were then averaged over these 5 runs. Conversely, to compute the perplexity score on excerpts from the Interview category one LM was acquired from the Rally class, and used to test on all 5 transcripts. The same procedure was followed for the training/fine-tuning on the Interview category: leave-one-out schema for testing on transcripts from the same class, and only one model to compute the perplexity of transcripts in the other class. Regarding the LMs acquired through GPT-2, fine-tuning was performed on the selected transcripts (different settings were tested, and finally 30 epochs and windows of 50 tokens were employed).

We then expected to observe analogous perplexity scores on all transcripts (as capturing common features underlying the language of the same speaker); and to observe slightly higher perplexity scores with models trained/fine-tuned on Interviews (Rallies) and used to test on Rallies (Interviews).

Results

The results are presented in Table 7.1, where we recorded the average perplexity scores and their standard deviation. We can see that perplexity scores range over a small interval on all considered sorts of LMs, thus confirming the main prediction that the training/fine-tuning on transcriptions from the same subject produces a language model that substantially grasps the main traits of that subject's language. The standard deviation-to-perplexity ratio is always lower than 10% (5.53% on average, through all conditions), thereby showing the reduced dispersion in these considered perplexity scores.

Additionally, the same scores seem to corroborate the second hypothesis, that the perplexity metrics is subtle enough to reflect the different contributions of the two sorts of language, employed in Interview vs. Rally. In fact, we can verify that by training on Interviews we obtain higher perplexity scores when testing on transcripts from the Rally category, and the same holds conversely by training on documents from the Rally category and testing on Interviews.

7.4.3 Experiment 2: Intra-subject coherence on different speakers

Also, the second experiment is aimed at assessing whether perplexity scores are stable within subject.

Five transcripts with no specific topic for eight well-known past and present political figures were selected and the associated perplexity scores and standard deviations were analyzed to confirm the results obtained in the previous experiment on a larger set of speakers.

Materials

In this case the context was less uniform than in the previous experiment, in that we collected political rallies, speeches on spot topics, such as economy, health systems, general challenges for the Western economy, a talk given in Davos, civil rights, and so forth. The complete list and URLs of the employed material is provided in A.3.4. Time duration, number of tokens and number of unique tokens describing the transcripts employed in this experiment are also presented in the Appendix, in Table A.11.

Procedure

A leave-one-out setup has been implemented, that is for each subject we employed 4 of the 5 transcripts for fine-tuning the GPT-2 base model and to acquire Bigrams, while the fifth transcript was used to compute the perplexity score. Regarding the experiment involving GPT-2, we devised a second trial in which a general LM with no specific fine-tuning has been employed to compute the perplexity for all considered transcripts. In both trials involving language models based on GPT-2 the training has been performed on 30 epochs, with window sized to 50 tokens.

Table 7.2: Perplexity scores along with their standard deviations of the experiment testing intra-subject coherence for the eight considered speakers. More specifically, we report the scores obtained by employing the Bigram model and the two versions of GPT-2, comparing the perplexity scores obtained through fine-tuned language models vs. the base GPT-2 model.

	Bigrams		GPT-2 Trial 1		GPT-2 Trial 2	
Subject	PPL	stdev	PPL	stdev	PPL	stdev
J. Biden	695.73	59.55	28.06	0.83	44.37	1.62
D. Trump	582.02	24.70	19.48	0.61	42.41	1.87
B. Obama	538.91	32.79	20.09	7.81	39.20	2.29
B. Sanders	605.71	38.59	22.34	5.67	29.91	3.99
B. Gates	462.64	38.59	34.83	6.08	43.31	6.01
N. Mandela	823.85	170.09	37.68	5.63	39.99	4.04
M. L. King	707.65	97.31	26.08	8.32	43.14	4.87
B. Johnson	442.03	44.45	50.59	10.78	63.90	12.12

The first part of the experiment (which we call *First Trial*) has been designed so to assess the perplexity scores featuring the language of each subject. In the *Second Trial* the GPT-2 base model with no fine-tuning was employed. The Second Trial is intended to measure the reliability of perplexity scores with respect to a language model reflecting a general language, not specifically tuned on the transcripts at hand.

In the first trial we expected to obtain scores varying in a small interval, under the assumption that if perplexity is appropriate to grasp the main linguistic traits of different speakers, different speeches from the same speaker may be featured by different scores, but with reduced standard deviation. Regarding the investigation of GPT-2 LMs, we expected to record higher perplexity scores —paired to higher standard deviation— in the second trial with respect to the first one, in which the LM has been fine-tuned in order to grasp the peculiar linguistic traits of each speaker.

Results

The results of the second experiment are reported in Table 7.2, which provides figures averaged over the 5 transcripts available for each subject (detailed results have been postponed to the Appendix, Table A.12).

Let us start by presenting the results obtained in the GPT-2 trials. The perplexity scores are in the same order of magnitude, with standard deviations indi-

cating a variable but mostly reduced dispersion. The LMs built through Bigrams are featured by an average 9.76 standard deviation-to-perplexity ratio across subjects, while those employing GPT-2 LMs range from 10.18 (Trial 2) to 19.49 (Trial 1). Comparing the perplexity scores obtained through LMs based on GPT-2 in Trials 1 and 2, we observe a clear increase in the average perplexity scores obtained in the second trial: the average across all subjects of perplexity scores grows from 30.33 up to 43.30. Such figures confirm that the GPT-2 with fine-tuning on the language from a given subject is able to specialize the LM thereby resulting more predictive for that subject's language. Furthermore, and perhaps more importantly in order to assess perplexity as a reliable metrics to analyze individuals'language, perplexity scores seem to capture intra-subject coherence. This datum holds for all (Bigrams and GPT-2) language models at stake.

Regarding the scores computed through Bigram-based LMs, we observe a clear increase in the absolute perplexity values, paired to a proportionally reduced standard deviation.

By looking at the data in Table 7.2 we cannot individuate a clear trend in the scores computed through LMs based on Bigrams and GPT-2: for example, PPL scores for Boris Johnson transcripts are the lowest ones when computed with models based on Bigrams, and highest ones when employing GPT-2 models. We thus decided to deepen the analysis by focusing on a subset of results, and analyzed the scores recorded for two subjects, J. Biden (JB) and B. Johnson (BJ). Transcripts for JB have been selected as exhibiting intermediate PPL and smallest standard deviation in the GPT-2 Trial 1, whilst the samples from BJ have been selected as showing the maximal difference between the two underlying LMs, Bigrams and GPT-2.

Biden vs. Johnson The scores recorded for the mentioned speakers are provided in full detail in Table 7.3.

Let us start by commenting perplexity scores computed through GPT-2 LMs: in the first case (JB) we obtained rather homogeneous perplexity scores (ranging between 27.43 and 29.42 in the first trial, and between 42.29 and 46.28 in the second trial), while in the second case (BJ) scores vary between 37.87 and 62.44 in the first trial, and between 53.89 and 83.04 in the second trial. By comparing JB and BJ scores, we observe that the linguistic features of JB are closer to the LMs acquired

Table 7.3: Perplexity scores (paired to standard deviation) associated to each transcript (column T) by Joe Biden (JB) and Boris Johnson (BJ: scores obtained through Bigrams, LSTM and GPT-2 are reported. In the second trial the GPT-2 base model (with no fine-tuning) was employed.

Subject	Test	Bigrams	GPT-2 Trial 1	GPT-2 Trial 2
J.B.	I	727.74	27.40	42.29
	II	653.82	27.31	43.59
	III	612.58	29.33	45.70
	IV	733.25	28.40	44.01
	V	751.25	27.87	46.28
B.J.	I	391.33	40.76	54.69
	II	418.57	37.87	53.89
	III	435.47	62.44	83.04
	IV	455.18	53.81	59.71
	V	509.61	58.05	68.19

by GPT-2: based on Trial 1 evidences one would argue that the considered speeches are more consistent and uniform in an intra-subject perspective, and based on Trial 2 one would argue that the language of JB is more consistent also with the general language model (as approximated by the base model). Many aspects may have contributed to this datum, such as the origin of texts employed to build the GPT-2 model, its bias in favor of American English, the fact that some topics (along with domain specific dictionary and idiomatic expressions) may be over- or under-represented, usage of coordination *vs.* subordination, lexical choice, and so forth. If we inspect the speeches by JB we realize that these are rather homogeneous, since their focus is largely on economic matters, plus a political rally and a discourse to Air Force Personnel:

- Biden I: drive-in campaign rally event in Cleveland, Ohio on November 2, 2020;
- Biden II: speech on 2020 job numbers and the state of the economy on December 4, 2020;
- Biden III: remarks to U.S. Air Force Personnel and Families Stationed at Royal Air Force Mildenhall, June 09, 2021;
- Biden IV: remarks on the economy, at the Coyahoga Community College in Cleveland, Ohio, May 27, 2021;
- Biden V: speech introducing his $2 trillion infrastructure plan on March 31,

2021.

On the other hand, the matters addressed by Boris Johnson are as varied as economics, public education program, anti-COVID plan, discourse to the UN general assembly, and the so-called 'levelling up', consisting of many initiatives, such as high street regeneration, local transport projects and cultural assets. This is the list of considered speeches.

- Johnson I: speech on the economy in Dudley on June 30, 2020;
- Johnson II: speech at Exeter College, England, on September 29, 2020; (topic: Lifetime Skills Guarantee to help people train and retrain at any stage in their lives)
- Johnson III: statement on the COVID-19 Winter Plan, on November 23, 2020;
- Johnson IV: remarks at the UN General Assembly, on September 24, 2019;
- Johnson V: speech on levelling up the UK, on July 15, 2021.

Such topics are not only diverse and different from each other, which explains the higher perplexity scores obtained in both GPT-2 trials, but also rather different from the topics on which the GPT-2 model has been trained (in particular COVID issues and levelling up activities, which were absent from the set of texts used to train GPT-2). This seems to be an argument in favor of the reliability of the perplexity metrics, which overall grows as the language being considered is less consistent with that used to train the models.

Based on the aforementioned arguments, we conclude that PPL as computed through GPT-2 is a reliable metrics. We still have to elaborate on why PPL values computed through Bigram models differ to such an extent from these. In fact, provided that we are mostly concerned with demonstrating the reliability of the perplexity in an intra-individual setting, nonetheless it may be useful to investigate the reasons why models based on Bigrams —differently from those based on GPT-2— produce higher perplexity values for JB transcripts than for those by BJ.

In order to elaborate on this question, it may be helpful to take into account some further statistics that may affect the behavior of Bigrams, that is: *i)* the ratio between unknown words (UNK) and all tokens in the training set; *ii)* the inverse of

Table 7.4: For each transcript from JB and BJ the ratio between unknown words (UNK) and overall number of tokens, the inverse of perplexity and standard deviation scores are reported.

Subject		UNK/tokens	PPL^{-1}	$stdev^{-1}$
J.B.	I	0.05	0.00137	0.00190
	II	0.05	0.00153	0.00220
	III	0.06	0.00163	0.00234
	IV	0.06	0.00136	0.00199
	V	0.05	0.00133	0.00185
B.J.	I	0.13	0.00256	0.00353
	II	0.12	0.00239	0.00362
	III	0.11	0.00230	0.00299
	IV	0.10	0.00220	0.00342
	V	0.12	0.00196	0.00278

PPL; and *iii)* the inverse of standard deviation. We recall that while building Bigram models tokens occurring only once in the training set were replaced with the tag UNK. Also, at testing time, OOVs (out-of-vocabulary words) were mapped onto the unknown word tag (please refer to Section 7.4.1). This strategy was developed to allow training Bigrams even with a small training set; however, it has the obvious drawback of substantially altering the probabilities associated to sequences involving tokens occurring only once in data. With this rewriting, tokens that are seldom seen in the training data become part of a class which instead is frequent at training time: in accord with intuition, N-grams involving tokens in this class are typically associated to low perplexity scores. To verify such intuition we reported in Table 7.4 the ratio between UNK and all tokens in the training set, along with the inverse of perplexity and standard deviations obtained through Bigram based models. We computed the Pearson correlation r between the ratio and the inverse of PPL scores, and obtained that $r(\text{UNK/tokens}, \text{PPL}^{-1}) = 0.59$. This mechanism may be helpful in explaining why transcripts by BJ, featured by a higher UNK/tokens ratio, obtained lower perplexity scores than JB (whose transcripts exhibit a smaller UNK/tokens ratio). A similar argument may explain the relation intervening between UNK/tokens ratio and inverse standard deviation, for which we recorded a Pearson correlation $r(\text{UNK/tokens}, \text{stdev}^{-1}) = 0.63$.

Also, if the growth of the UNK/tokens ratio is detrimental to the perplexity because it hinders the appropriate estimation of Bigrams involving UNK tokens, one can argue that conversely, Bigrams may be appropriate to cope with data sets featured by few UNK tokens. This hypothesis is also tested in the experiment described in the following Section.

7.4.4 Experiment 3: Predictive and discriminative features of PPL

For this experiment we used publicly available data from the Pitt Corpus.[5] These data were gathered as part of a larger protocol administered by the Alzheimer and Related Dementias Study at the University of Pittsburgh School of Medicine (Becker et al., 1994). In particular, we selected the descriptions provided to the Cookie Theft picture, which is a popular test used by speech-language pathologists to assess expository discourse in subjects with disorders such as dementia. This dataset was employed to test whether perplexity scores on the collected descriptions allow discriminating patients from healthy controls.

Materials

The dataset is composed of 552 files arranged into Control (243 items) and Dementia (309 items) directories. These correspond to multiple interviews to 99 control subjects, and to 219 subjects with dementia diagnosis. Text documents herein were transcribed according to the CHAT format,[6] so we pre-processed such documents to extract text. In so doing, the original text was to some extent simplified: e.g., pauses were disregarded, like hesitation phenomena, that were not consistently annotated (MacWhinney, 2014, 2017). In particular lengthened syllables, long pauses and interruption symbols were eliminated, alongside a wide variety of sounds such as cries, sneezes, and coughs. Other meaningful aspects were preserved in the final file, such as repetitions, interjections and retracings, considering these events as important features for the model to capture. No information on intonational contours and other markers of the utterance planning process was available in the input files.

To the ends of collecting enough text to be analyzed, we dropped the interviews of subjects that participated in only one interview. We ended up with material relative to 74 control subjects (for which overall 218 transcripts were collected), and to 77 subjects with dementia diagnosis (overall 192 transcripts).

[5] https://dementia.talkbank.org/access/English/Pitt.html.
[6] https://talkbank.org/manuals/CHAT.pdf.

Procedure

This experiment is aimed at discriminating subjects tagged as belonging to the Alzheimer's disease (AD) class from healthy controls. Intuitively, such categorization has been performed as follows. The LMs employed to compute perplexity for each transcript were trained on healthy subjects. Since it is largely acknowledged in literature that the language of patients affected by dementia significantly differs from that of healthy subjects, we formulated the hypothesis that models acquired on healthy subjects' language will better predict the language of healthy subjects than the language of subjects affected by dementia. Put in other words, we expected lower perplexity scores to be associated to transcripts from healthy subjects, whilst transcripts from dementia patients to result in higher perplexity scores.

Since, as mentioned, the models were trained on healthy controls to recognize non-healthy patients, we devised a twofold procedure to compute the average perplexity on control subjects. (i) In order to classify subjects in the class of healthy controls, the training of the LM was performed in a leave-one-subject-out setting. Namely, language models have been refined with files from all other control subjects except for one, which has been used for testing. Conversely, only one model acquired on healthy control was necessary to compute the perplexity associated to all subjects in the AD class. To compute the classification, we exploited the average perplexity scores characterizing all control transcripts as threshold for the decision rule. (ii) In order to classify subjects in the AD class, a single model was acquired on all control transcripts so as to compute the average perplexity score used as threshold.

As *decision rule* to discriminate AD patients from healthy subjects we used the average perplexity scores characterizing all control transcripts employed in the training process as our threshold. In case the perplexity score averaged on all available transcripts for a given subject was higher than the average of healthy controls, we marked the subject as suffering from AD; as healthy otherwise.

A twofold experimental setting has been devised, including experiments with Bigrams and GPT-2, adopting a window size set to 20 in order to handle shorter text samples. In the case of GPT-2, the model has been trained for 30 epochs.

Table 7.5: Results of the experiments in the third experiment: more specifically, we compare the categorization results obtained with LMs acquired through Bigrams and GPT-2 by reporting Accuracy, Precision (P), Recall (R) and F1 scores.

Class	Bigrams				GPT-2			
	Acc.	P	R	F1	Acc.	P	R	F1
Dementia	59.60%	0.56	0.94	0.70	66.23%	0.61	0.92	0.73
Control		0.78	0.24	0.37		0.82	0.39	0.53

Evaluation Metrics

To evaluate the results in the third experiment we adopted the Precision and Recall metrics (specificity and sensitivity) along with their harmonic mean, F1 score, and accuracy. Precision (specificity) is defined as $P = \frac{TP}{TP+FP}$, while Recall (sensitivity) is defined as $R = \frac{TP}{TP+FN}$. Informally stated, Precision computes the fraction of results that are actually correct: it is computed as the number of correct results (true positives, TP) divided by the sum of correct results (TP) and items mistakenly returned as results (false positives, FP). Recall computes how many correct results were individuated. In Recall, we have the number of correct results divided by the sum of correct results (TP) and items mistakenly not recognized as results (false negative, FN). While precision provides an estimation of how precise a categorization system is, recall indicates how many results were identified out of all the possible ones. F_1 measure is then used to provide a synthetic value of Precision and Recall, whereby the two measures are evenly weighted through their harmonic mean: $F_1 = 2 \cdot \frac{P \cdot R}{P+R}$

Accuracy is computed as $ACC = \frac{TP+TN}{P+N}$, that is as the fraction of correct predictions (the sum of TP and TN) over the total number of records examined (the sum of positives and negatives, P and N).

Results

The results obtained on discriminating AD patients from controls are provided in Table 7.5: the categorization employing GPT-2 obtained .73 and .53 F1 score on the Dementia and Control class, respectively. Recorded accuracy was in the order of 66.23%. Slightly lower figures were obtained with LMs acquired through Bigrams, ranging from .70 and .37 F1 score on the Dementia and Control class, respectively; accuracy was 59.60%.

Such figures also show that, independent from the complexity of the adopted model, simple categorization algorithms based on perplexity scores can achieve valuable results, provided that the employed LMs are trained on a consistent transcript category. This datum shows that systems implementing different sorts of LMs can be helpful for practical uses, as a tool for assisting specialists in the diagnostic process.

8 Conclusions

In this work we discussed the relevance of a semantic layer on top of language models: tying language models to a symbolic knowledge representation allows modeling semantic characterizations sharing the space defined by such models. We therefore illustrated the motivations that have brought to the development of two novel lexical resources: LESSLEX and SE-MACAROON. Evaluating such resources allowed us to investigate the role played by semantic information when dealing with the semantic similarity task, by also making explicit a new latent task, the sense identification task. On a different perspective, we subsequently evaluated language models in an application setting: we tested whether and to what extent language models can be exploited as linguistic analysis devices.

In Chapter 4 we have proposed LESSLEX vectors. The research question answered herein is focused on how to integrate symbolic knowledge with distributional resources to build sense embeddings. Such vectors are built by re-arranging distributional descriptions around senses, rather than terms. These have been tested on the word similarity task, on the contextual similarity task, and on the semantic text similarity task, providing good to outstanding results, on all datasets employed. We have discussed the obtained results. Also importantly, we have outlined the relevance of LESSLEX vectors in the broader context of research in natural language with focus on senses and conceptual representation, mentioning that having co-located sense and term representations may be helpful to investigate some issues in an area at the intersection of general AI, Cognitive Science, Cognitive Psychology, Knowledge Representation and, of course, Computational Linguistics. In these settings distributed representation of senses may be employed, either to enable further research or to solve specific tasks. The conceptual grounding on BabelNet enables LESSLEX dealing with the 284 different languages (provided by BabelNet version 4.0). It also enables LESSLEX vectors to be plugged into applications

that already adopt such sense inventory. Differently from most sense embedding approaches, LESSLEX exploits the feature of adopting a unique semantic space for concepts and terms from different languages. Far from being an implementation feature, the adopted semantic space describes a cognitively plausible space, compatible with the cognitive mechanisms governing lexical access, which is in general featured by conceptual mediation (Marconi, 1997). Such feature allowed us to compare and unveil meaning connections between terms across different languages. Such capabilities can be useful in characterising subtle and elusive meaning shift phenomena, such as diachronic sense modeling (Hu, Li, & Liang, 2019) and conceptual misalignment, which is a well-known issue, e.g., in the context of automatic translation. This issue has been approached, for the translation of European laws, through the design of formal ontologies (Ajani et al., 2010).

We also proposed a novel semantic similarity measure, the ranked-similarity. Such novel measure originates from a simple intuition: in computing conceptual similarity, scanning and comparing each and every sense available in some fine-grained sense inventory may be unnecessary and confusing. Instead, we rank senses using their distance from the term; top ranked senses are more relevant, so that the formula to compute ranked-similarity refines cosine similarity by adding a mechanism for filtering and clustering senses based on their salience. Acquiring vector descriptions for concepts enables to investigate the conceptual abstractness/concreteness that has recently emerged as central in the multidisciplinary debate between grounded views of cognition versus modal (or symbolic) views of cognition (Colla, Mensa, Porporato, & Radicioni, 2018; Hill, Korhonen, & Bentz, 2014; Mensa, Porporato, & Radicioni, 2018). Also accounting for conceptual abstractness may be beneficial in diverse NLP tasks, like WSD (Kwong, 2008), the semantic processing of figurative uses of language (Neuman et al., 2013; Turney, Neuman, Assaf, & Cohen, 2011), automatic translation and simplification (Zhu, Bernhard, & Gurevych, 2010), the processing of social tagging information (Benz, Körner, Hotho, Stumme, & Strohmaier, 2011), and many others, as well.

In Chapter 5 we addressed two main research questions: first, we have proposed two metrics to compute semantic similarity involving sense embeddings, leveraging the different vectorial representation of senses and terms that most sense

embedding provide to date; additionally, we posited the sense identification task as a relevant complement to the semantic similarity task. The proposed metrics build on the intuition that the maximization ordinarily adopted as distance metrics can be refined by accounting for the centrality of the term representation with respect to the sense representations (in the case of the $\mathcal{R}$-sim metrics), and even more by gathering close senses (in the case of the $\mathcal{N}$-sim metrics). Different from the semantic similarity task, the sense identification requires to make explicit the senses underlying the similarity ratings.

We have sense-annotated a popular dataset for semantic similarity, and used it for experimentation, to assess the proposed metrics. Our experiments investigated how the novel metrics fit to the specific features of an array of six recent vectorial resources in both the considered tasks, semantic similarity and sense identification. The experimentation revealed that ranked similarity ($\mathcal{R}$-sim) and neighbourhood similarity ($\mathcal{N}$-sim) mostly allow obtaining more accurate results also in the semantic similarity task; and they are never worse than the familiar maximization of cosine similarity ($\mathcal{M}$-sim). Systems employing sense embeddings can thus simply replace $\mathcal{M}$-sim with $\mathcal{N}$-sim. As illustrated in the discussion, in the worst scenario this would lead to minimal improvement in the semantic similarity correlation, but also to a consistent gain in the sense identification. Our experimental evidences seem to suggest that the resources featured by sense level indexing will be useful mostly for tasks dealing with semantic similarity, where more specific and covering vector representations are needed, while the less covering but more precise term-sense indexing seems more appropriate to target tasks where sense individuation is the primary concern.

In Chapter 6 we have proposed SE-MACAROON vectors. Such vectors are built by collecting contextualized descriptions of words as expression of a given word sense. The third research question underlying this study was concerned with the role played by contextual vector descriptions in representing word senses. The effectiveness of such representations was successfully tested on the Word Sense Disambiguation task. SE-MACAROON vectors have been tested on the Word Sense Disambiguation task, providing results comparable to the state-of-the-art contextualized sense embeddings. We have discussed the obtained results, also

presenting a novel WSD approach enjoying both the shared space between senses and terms as well as for the peculiar sense representations. Different from the proposed contextualized sense embedding techniques, SE-MACAROON represents word senses as collections of word embeddings rather than conflating all of its occurrences into a unique representation. Such feature provides the resource with a sort of lexical memory, storing the ideal representations of a word sense taken in context, thus providing sense representation close to several context of usage. This allowed us to devise a novel approach to WSD, exploiting multiple occurrences for each word sense. The building rationale behind SE-MACAROON vectors stems from the hypothesis that aggregating multiple representations of the same word sense into a single one might end up with misleading or imprecise information. We therefore investigated the impact of limiting the number of occurrences stored for each word sense. The results of such experiments seem to support our hypothesis: limiting the memory size to few occurrences seems to be detrimental to performances; conversely, fixing the limit to 100 seems to lead to performances comparable to the unbounded setting. In addition to the peculiarity of SE-MACAROON representations, the proposed WSD approach relies on the meaning localization principle: in order to determine the sense underlying a word we may account for the local context rather than considering the entire sentence. We then investigated the impact of parameters of the WSD approach. The results corroborate the hypothesis that a small context window surrounding the target word provides helpful information while solving the WSD task. In particular, as the window size increases, performance drops systematically.

In Chapter 7 we investigated whether the perplexity metrics can be interpreted as a semantic coherence marker, thereby allowing employing language models in the early detection of psychotic disorders. After having presented two resources, in this chapter we show how to employ an intrinsic metrics (originally concerned with evaluating the 'fit' of a language model to actual language) to predict the insurgence of a broad class of cognitive impairments affecting linguistic production. The diagnosis of dementia is a complex process that is long and labor intensive, involving a neuropsychiatric evaluation that includes medical and neurologic history and examination, semistructured psychiatric interview, and neuropsychological as-

sessments (Huff et al., 1987; Lopez et al., 1990). To this extent, we have been exploring whether perplexity can be considered as a reliable metrics to analyze spoken language at large. To answer this question we designed an experiment to compare perplexity scores for different speeches from the same speaker (transcripts from an healthy subject were considered in this phase): two sorts of language —political rallies and interviews— were analyzed. In the second experiment we compared perplexity scores associated to the language of eight past and present popular political figures. The results of these studies seem to corroborate the hypothesis that perplexity can be measured in a reliable manner for the individual subject, while at the same time accounting for different linguistic registers. Differences in scores obtained through the application of different language models were detected and discussed. We observed that the perplexity computed through simpler LMs may be a good option when either language variability is reduced or training data ensure good coverage for the considered language. Conversely, simpler models may be misled by out-of-vocabulary terms: interestingly enough, however, even in these cases perplexity scores were consistent with the individual subject language characteristics.

Both LESSLEX and SE-MACAROON constitute an effort to build a semantic layer on top of language modes. The usability of distributional representations paired with more precise symbolic knowledge of semantic networks represents a complementary and yet interoperable combination of information, thus resulting in such sense embeddings. Under a different perspective, LESSLEX and SE-MACAROON are fully-fledged language models. This feature allows both LESSLEX and SE-MACAROON to host conceptual descriptions along with word representations, thus enabling to investigate new approaches combining both representations.

In the last few years distributional representations and neural networks have shaken up the NLP landscape (C. D. Manning, 2015). The introduction of transformers architecture started a whole host of experiments as well as for a new beginning for language models. In such a framework predicting directions for the future is not simple nor clear; we can foresee, however, some major aspects. First, contextualized language models have been assumed as standard *de facto*, proving their generalization capabilities across many different tasks as well as for different

domains. Such direction will be investigated in depth, to the ends of both building larger pre-trained models and building models capable of addressing many different domains and tasks avoiding pre-training process. Du et al. (2021) is one example in the latter direction, larger models may be provided with dedicated components for each different task or domain. The shift to the multi-task setting and towards transfer learning frameworks has already been started by impressive models, such as (Aribandi et al., 2021). In such a frame, the ability to deal with many languages at a time will be crucial: encoding information in many diverse languages allows accounting for the representation of the same item across languages and addressing different tasks in any language with the same model. In these respects, despite such models proved outstanding abilities in few shot learning, the lack of data in the lexical semantics area is still a serious concern. We may expect to assist a new wave of datasets for many different NLP tasks. However, manually annotating data is still difficult and time-consuming, especially when dealing with word senses, this represents one of the chief limitations in such a rapidly changing landscape. This datum is becomes apparent if we consider that the main project on manually sense-annotated data is SemCor, developed decades ago, in 1994.

A second relevant line of research will be, in our view, the attention to the evaluation frameworks of such models: the rapid improvements of neural models have overcome the ability of many benchmarks to assess them. We may expect to see even larger benchmark development. In this respect, recent questions on the best practices to assess such models have emerged. In our view, the community will face the issue of assessing whether models are learning to address benchmarks instead of tasks. Interpretability and explainability of these models deserve to be mentioned, and will likely play an increasingly relevant role. Dealing with dense representations, together with large models, makes systems decisions rather opaque. In this setting, representing word senses may provide beneficial effects in making interpretable such decisions, by making explicit the semantic grounding around which information can be contextualized.

Finally, next steps will involve combining word senses: compositions based on syntactic information, or based on different principles. For example, such syntactically-enriched representations will be employed in representing and recognizing events,

and in modeling the relationships that tie them together. For example, given news articles, we may want to recognize the sequence of events, along with their participants, of an earthquake.

The work done all throughout my PhD course, summarized in this thesis, is to a good extent linked to all these directions. LESSLEX and SE-MACAROON adopt sense-oriented representations, that hopefully will allow coping with all mentioned challenges.

A Appendix

A.1 Results on the word similarity task, CbA condition

In this Section we illustrate the results obtained by testing on the semantic similarity task. However, different from the results reported in Section 4.2.1, in this case only the fraction of each dataset covered by all considered resources was used for testing.

Table A.1: Results on the subset of the multilingual and cross-lingual RG-65 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

RG-65	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
[Word] eng [1.0]	.64	.59	**.91**	.86	**.91**	**.90**	.67	.67	.84	.86	.75	.81	.80	.75
[Sense] eng [1.0]	-	-	**.94**	**.91**	-	-	.81	.76	-	-	.72	.76	.78	.73
fas (N) [.69]	.78	.73	.86	.87	**.88**	**.89**	.71	.69	-	-	.72	.60	-	-
spa (N) [.98]	.82	.82	**.92**	**.93**	**.92**	**.93**	.91	.91	.80	.83	.82	.84	-	-
por-fas (N) [.81]	.73	.72	.91	**.90**	**.93**	.89	.79	.76	-	-	.76	.70	-	-
fra-por (N) [.97]	.83	.84	**.93**	**.89**	**.93**	**.89**	.76	.69	-	-	.81	.73	-	-
fra-fas (N) [.87]	.72	.72	.90	.88	**.93**	**.89**	.73	.69	-	-	.74	.68	-	-
fra-spa (N) [.99]	.81	.80	**.93**	**.91**	**.93**	.89	.85	.83	-	-	.88	.86	-	-
fra-deu (N) [.99]	.82	.86	**.91**	**.90**	.89	.88	.81	.78	-	-	.78	.76	-	-
spa-por (N) [.98]	.83	.83	**.93**	**.92**	**.93**	**.92**	.83	.81	-	-	.80	.79	-	-
spa-fas (N) [.82]	.71	.69	.92	**.92**	**.93**	.91	.83	.82	-	-	.78	.83	-	-
eng-por (N) [.99]	.74	.72	**.94**	**.90**	.92	**.90**	.79	.76	-	-	.80	.77	-	-
eng-fas (N) [.83]	.68	.61	.92	.89	**.93**	**.92**	.79	.74	-	-	.78	.74	-	-
eng-fra (N) [1.0]	.71	.70	**.94**	**.92**	.92	.91	.76	.73	-	-	.81	.75	-	-
eng-spa (N) [.99]	.73	.71	**.93**	**.93**	**.93**	.92	.85	.85	.84	.85	.80	.85	-	-
eng-deu (N) [.98]	.74	.72	**.92**	**.90**	.90	**.90**	.83	.81	-	-	.77	.80	-	-
deu-por (N) [.96]	.89	.86	**.93**	**.89**	.92	.88	.82	.78	-	-	.77	.74	-	-
deu-fas (N) [.81]	.76	.74	**.92**	**.91**	**.92**	.90	.88	.81	-	-	.82	.82	-	-
deu-spa (N) [.97]	.85	.86	**.92**	**.91**	.91	.90	.89	.86	-	-	.80	.81	-	-

Table A.2: Results on the subset of the WS-Sim-353 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

WS-Sim-353	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N) [.97]	.67	.65	**.78**	**.79**	**.78**	**.79**	.60	.61	.75	.76	.69	.73	.71	.70
ita (N) [.92]	.68	.69	.74	**.77**	**.75**	**.77**	.66	.65	.69	.70	.65	.71	-	-
deu (N) [.88]	.77	.74	.83	.81	**.84**	**.83**	.70	.69	-	-	.65	.64	-	-
rus (N) [.83]	.75	.76	.77	.78	**.79**	**.79**	.66	.66	-	-	.63	.64	-	-

Table A.3: Results on the subset of the SimVerbs-3500 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

SimVerbs-3500	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (V) [1.0]	.58	.56	**.67**	**.66**	.62	.60	-	-	.56	.56	.45	.42	.31	.30

A.2 Results on the Sense Identification Dataset

A.3 Sources of experimental material and detailed results

A.3.1 Material used in Experiment 1

URLs of transcripts in the class Interview:

- https://www.rev.com/blog/transcripts/full-transcript-of-donald-trump-interview-with-meet-the-press
- https://www.rev.com/blog/transcripts/donald-trump-unedited-60-minutes-interview-transcript
- https://www.rev.com/blog/transcripts/donald-trump-rush-limbaugh-interview-radio-rally-transcript-october-9
- https://www.rev.com/blog/transcripts/donald-trump-interview-transcript-with-axios-on-hbo
- https://www.rev.com/blog/transcripts/donald-trump-election-day-interview-transcript-fox-friends

URLs of transcripts in the class Rally:

- https://www.rev.com/blog/transcripts/donald-trump-gastonia-nc-rally-speech-transcript-october-21

Table A.4: Results on the subset of the multilingual SimLex-999 containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

SimLex-999	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N) [1.0]	.51	.52	**.69**	**.67**	.66	.63	.41	.39	.55	.53	.52	.49	.46	.44
eng (V) [1.0]	.62	.56	**.67**	**.65**	.61	.58	-	-	.51	.50	.54	.49	-	-
eng (A) [1.0]	**.84**	**.83**	.82	.79	.80	.78	-	-	.63	.62	.55	.51	-	-
eng (*) [1.0]	.57	.53	**.70**	**.69**	.67	.65	-	-	.55	.54	.53	.49	-	-
ita (N) [.96]	.50	.49	**.66**	**.64**	.64	.62	.48	.49	.48	.49	.56	.50	-	-
ita (V) [.96]	.58	.53	**.70**	**.63**	.69	.59	-	-	.57	.50	.56	.45	-	-
ita (A) [.95]	.68	.57	**.77**	**.70**	.73	.64	-	-	.40	.30	.61	.49	-	-
ita (*) [.96]	.49	.43	**.67**	**.63**	.65	.62	-	-	.48	.46	.55	.48	-	-
deu (N) [.94]	.58	.57	.66	.65	**.68**	**.66**	.46	.47	-	-	.48	.44	-	-
deu (V) [.73]	.56	.53	.63	**.60**	**.64**	.58	-	-	-	-	.51	.46	-	-
deu (A) [.67]	.74	.70	.76	.73	**.80**	**.75**	-	-	-	-	.50	.39	-	-
deu (*) [.86]	.59	.57	.66	.65	**.69**	**.67**	-	-	-	-	.47	.42	-	-
rus (N) [.86]	.45	.43	**.54**	**.51**	**.54**	.49	.23	.23	-	-	.26	.21	-	-
rus (V) [.20]	.60	.54	.58	.59	**.66**	**.60**	-	-	-	-	.42	.28	-	-
rus (A) [.06]	.92	.87	**.94**	**.91**	**.94**	.87	-	-	-	-	.62	.24	-	-
rus (*) [.63]	.46	.44	**.55**	**.51**	**.55**	.50	-	-	-	-	.27	.21	-	-

- https://www.rev.com/blog/transcripts/donald-trump-rally-transcript-tucson-arizona-october-19

- https://www.rev.com/blog/transcripts/donald-trump-macon-georgia-rally-speech-transcript-october-16

- https://www.rev.com/blog/transcripts/donald-trump-middletown-pa-rally-speech-transcript-sept-26-first-rally-after-scotus-nomination

- https://www.rev.com/blog/transcripts/donald-trump-newport-news-virginia-campaign-rally-transcript-september-25

Table A.5: Results on the subset of the SemEval 17 Task 2 dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

SemEval-2017	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
eng (N) [.66]	.70	.70	**.84**	**.86**	.83	.85	.57	.59	.75	.77	.71	.75	.73	.73
deu (N) [.73]	.78	.79	**.84**	.85	**.84**	**.86**	.68	.68	-	-	.67	.69	-	-
ita (N) [.61]	.73	.73	**.82**	**.84**	.80	.82	.75	.76	.76	.78	.71	.77	-	-
spa (N) [.62]	.77	.79	**.84**	**.86**	.81	.84	.70	.71	.78	.80	.73	.78	-	-
fas (N) [.34]	.69	.72	**.79**	**.82**	.75	.80	.58	.59	-	-	.65	.70	-	-
deu-spa (N) [.73]	.78	.80	**.84**	**.86**	.82	.84	.71	.72	-	-	.70	.74	-	-
deu-ita (N) [.74]	.77	.78	**.83**	**.85**	.82	.84	.72	.73	-	-	.69	.73	-	-
eng-deu (N) [.82]	.78	.79	**.85**	**.86**	.83	.85	.67	.68	-	-	.70	.72	-	-
eng-spa (N) [.63]	.74	.75	**.85**	**.87**	.83	.85	.65	.66	.75	.78	.72	.77	-	-
eng-ita (N) [.62]	.73	.74	**.85**	**.87**	.83	.85	.69	.70	.73	.75	.72	.77	-	-
spa-ita (N) [.61]	.75	.76	**.84**	**.86**	.81	.84	.74	.74	.70	.71	.72	.78	-	-
deu-fas (N) [.49]	.75	.78	**.84**	**.86**	.81	.85	.71	.72	-	-	.69	.74	-	-
spa-fas (N) [.49]	.72	.74	**.84**	**.86**	.80	.84	.70	.72	-	-	.70	.77	-	-
fas-ita (N) [.49]	.71	.72	**.81**	**.84**	.72	.82	.70	.72	-	-	.69	.75	-	-
eng-fas (N) [.54]	.70	.71	**.82**	**.85**	.79	.82	.65	.68	-	-	.70	.75	-	-

A.3.2 Statistics of the data employed in Experiment 1

Table A.6: Results on the subset of the Goikoetxea dataset containing only word pairs covered by all considered resources. Reported figures express Pearson (r) and Spearman (ρ) correlations. In the first column we report the coverage for each translation of the dataset actually used in the experimentation.

Goikoetxea	LL-M		LLX		CNN		NAS		JCH		SSE		N2V	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
spa-eus (N) [.75]	.75	.71	.80	**.74**	**.81**	.73	.74	.73	.69	.66	.74	.70	-	-
eng-eus (N) [.77]	.75	.72	**.93**	**.91**	**.93**	.90	.91	.90	.87	.84	.84	.86	-	-
eng-spa (N) [.99]	.73	.71	**.93**	**.93**	**.93**	.92	.85	.85	.84	.85	.80	.85	-	-
eus-ita (N) [.72]	.62	.66	**.69**	**.73**	.67	.63	.57	.59	.58	.63	.53	.56	-	-
spa-ita (N) [.93]	.60	.65	**.67**	**.75**	.66	.74	.58	.59	.56	.61	.53	.59	-	-
spa-eus (N) [.73]	.67	.70	**.74**	**.79**	.71	.78	.66	.67	.70	.74	.60	.64	-	-
eng-ita (N) [.96]	.59	.64	**.70**	.76	**.70**	**.77**	.51	.52	.61	.66	.51	.58	-	-
eng-eus (N) [.75]	.64	.67	**.75**	**.80**	.74	**.80**	.58	.60	.72	.76	.58	.63	-	-
eng-spa (N) [.97]	.62	.66	**.72**	**.78**	.71	**.78**	.55	.56	.68	.74	.57	.64	-	-
eng-spa (N) [.97]	.50	.49	**.67**	**.65**	.64	.62	.52	.51	.56	.52	.55	.52	-	-
eng-spa (V) [.96]	.53	.49	**.62**	**.60**	.59	.57	-	-	.48	.46	.53	.49	-	-
eng-spa (A) [.80]	.76	**.77**	**.77**	**.77**	**.77**	**.77**	-	-	.59	.60	.56	.50	-	-
eng-spa (*) [.95]	.54	.52	**.67**	**.66**	.65	.64	-	-	.54	.52	.55	.51	-	-
eng-ita (N) [.97]	.53	.53	**.71**	**.69**	.68	.66	.46	.47	.53	.51	.55	.52	-	-
eng-ita (V) [.58]	.62	.55	**.71**	**.67**	.67	.60	-	-	.51	.45	.56	.46	-	-
eng-ita (A) [.80]	.79	.73	**.84**	**.78**	.78	.70	-	-	.41	.36	.61	.48	-	-
eng-ita (*) [.82]	.56	.53	**.72**	**.70**	.69	.67	-	-	.50	.48	.56	.50	-	-
spa-ita (N) [.96]	.53	.53	**.68**	**.67**	.66	.65	.47	.49	.48	.47	.56	.54	-	-
spa-ita (V) [.56]	.56	.52	**.65**	**.60**	.64	.58	-	-	.47	.42	.56	.49	-	-
spa-ita (A) [.78]	.73	.66	**.79**	**.73**	.76	.69	-	-	.43	.38	.63	.51	-	-
spa-ita (*) [.80]	.55	.53	**.68**	**.66**	.67	.65	-	-	.47	.45	.56	.51	-	-

A.3.3 Detailed perplexity scores obtained in Experiment 1

Table A.7: Results on the subset of SemEval 17 English dataset containing only the 213 term pairs covered by all the employed resources. Reported figures express Pearson (r), Spearman (ρ) correlations and their F1 score, and Precision (P) and Recall (R) along with their F1 score.

Resource	Measure	Semantic Similarity			Sense Identification		
		ρ	r	$F_1(\rho,r)$	P	R	F_1(P,R)
LLX	$\mathcal{M}$-sim	0.79	0.79	0.79	0.31	0.28	0.29
	$\mathcal{R}$-sim	0.87	0.86	0.86	0.50	0.46	0.48
	$\mathcal{N}$-sim	0.87	0.85	0.86	0.48	0.86	0.62
N2V	$\mathcal{M}$-sim	0.68	0.66	0.67	0.62	0.66	0.64
	$\mathcal{R}$-sim	0.76	0.76	0.76	0.63	0.58	0.60
	$\mathcal{N}$-sim	0.66	0.65	0.65	0.60	0.65	0.62
DCF	$\mathcal{M}$-sim	0.81	0.80	0.80	0.65	0.61	0.63
	$\mathcal{R}$-sim	0.81	0.80	0.80	0.73	0.68	0.70
	$\mathcal{N}$-sim	0.77	0.76	0.76	0.87	0.84	0.86
SSE	$\mathcal{M}$-sim	0.75	0.72	0.73	0.77	0.72	0.74
	$\mathcal{R}$-sim	0.76	0.74	0.75	0.88	0.83	0.85
	$\mathcal{N}$-sim	0.77	0.76	0.76	0.88	0.84	0.86
SW2V	$\mathcal{M}$-sim	0.78	0.77	0.77	0.73	0.68	0.70
	$\mathcal{R}$-sim	0.75	0.75	0.75	0.85	0.78	0.81
	$\mathcal{N}$-sim	0.77	0.77	0.77	0.85	0.81	0.83
$\text{LSTMBD}_{\text{T,S}}$	$\mathcal{M}$-sim	0.69	0.68	0.68	0.81	0.76	0.78
	$\mathcal{R}$-sim	0.70	0.69	0.69	0.89	0.83	0.86
	$\mathcal{N}$-sim	0.71	0.70	0.70	0.88	0.84	0.86
LSTMBD_{S}	$\mathcal{M}$-sim	0.77	0.74	0.75	0.68	0.63	0.65
	$\mathcal{R}$-sim	0.77	0.75	0.76	0.86	0.80	0.83
	$\mathcal{N}$-sim	0.79	0.77	0.78	0.82	0.82	0.82

A.3.4 Material used in Experiment 2

URLs of transcripts for Joe Biden:

- https://www.rev.com/blog/transcripts/joe-biden-drive-in-rally-speech-transcript-cleveland-november-2
- https://www.rev.com/blog/transcripts/joe-biden-speech-on-2020-job-numbers-economy-transcript-december-4
- https://www.whitehouse.gov/briefing-room/speeches-remarks/2021/06/09/remarks-by-president-biden-to-u-s-air-force-personnel-and-families-stationed-at-royal-air-force-mildenhall/
- https://www.c-span.org/video/?512149-1/president-biden-delivers-remarks-economy
- https://www.rev.com/blog/transcripts/joe-biden-speech-on-2-trillion-infrastructure-plan-transcript-march-31

Table A.8: Experiment considering only the 213 pairs of the SemEval 17 English dataset covered by all resources. Stats describing the number of senses available for each resource, along with the size of the neighborhood employed in the $\mathcal{N}$-sim metrics.

Resource	Measure	AVG term senses	AVG $\lvert\hat{S}\rvert$	$F_1(\rho,r)$	F_1(P,R)
LLX	$\mathcal{M}$-sim	16.40	3.65	0.79	0.29
	$\mathcal{R}$-sim			0.86	0.48
	$\mathcal{N}$-sim			0.86	0.62
N2V	$\mathcal{M}$-sim	13.47	1.30	0.67	0.64
	$\mathcal{R}$-sim			0.76	0.60
	$\mathcal{N}$-sim			0.65	0.62
DCF	$\mathcal{M}$-sim	3.75	1.09	0.80	0.63
	$\mathcal{R}$-sim			0.80	0.70
	$\mathcal{N}$-sim			0.76	0.86
SSE	$\mathcal{M}$-sim	5.11	1.07	0.73	0.74
	$\mathcal{R}$-sim			0.75	0.85
	$\mathcal{N}$-sim			0.76	0.86
SW2V	$\mathcal{M}$-sim	4.71	1.05	0.77	0.70
	$\mathcal{R}$-sim			0.75	0.81
	$\mathcal{N}$-sim			0.77	0.83
$\text{LSTMBD}_{\text{T,S}}$	$\mathcal{M}$-sim	2.39	1.05	0.68	0.78
	$\mathcal{R}$-sim			0.69	0.86
	$\mathcal{N}$-sim			0.70	0.86
LSTMBD_{S}	$\mathcal{M}$-sim	5.02	1.20	0.75	0.65
	$\mathcal{R}$-sim			0.76	0.83
	$\mathcal{N}$-sim			0.78	0.82

Table A.9: Statistics describing the transcripts employed in Experiment 1: for all considered samples we report time duration, the number of tokens, and the number of unique tokens.

Category	Transcript	Duration	# Tokens	# Unique Tokens
Rally	I	1 : 28 : 52	7,278	1,098
	II	1 : 28 : 23	6,471	922
	III	1 : 31 : 34	18,514	1,926
	IV	0 : 45 : 40	6,702	1,032
	V	1 : 01 : 51	5,933	946
Interview	I	1 : 17 : 37	15,200	1,967
	II	0 : 56 : 17	10,501	1,614
	III	1 : 43 : 43	20,865	2,300
	IV	1 : 13 : 01	14,056	1,945
	V	1 : 18 : 19	14,806	1,896

URLs of transcripts for Bill Gates:

- https://news.harvard.edu/gazette/story/2007/06/remarks-of-bill-gates-harvard-commencement-2007/

- https://www.gatesfoundation.org/ideas/speeches/2018/11/remarks-to-the-japanese-parliament

- https://prorhetoric.com/on-the-cusp-of-a-sanitation-revolution/

- https://www.gatesfoundation.org/ideas/speeches/2018/10/grand-challenges-annual-meeting

- https://www.gatesfoundation.org/ideas/speeches/2020/02/bill-gates-american-association-for-the-advancement-of-science

URLs of transcripts for Boris Johnson:

- https://www.gov.uk/government/speeches/pm-economy-speech-30-june-2020

- https://www.rev.com/blog/transcripts/boris-johnson-adult-education-training-speech-transcript-september-29

- https://www.gov.uk/government/speeches/pm-statement-on-covid-19-winter-plan-23-november-2020

- https://www.wired.com/beyond-the-beyond/2019/09/transcript-boris-johnsons-remarks-un-general-assembly/

- https://www.conservatives.com/news/levelling-up-speech-july-15

URLs of transcripts for Martin Luther King:

Table A.10: Perplexity scores obtained with fine-tuning on the transcripts from the Rally and Interview employed in Experiment 1 (Section 7.4.2).

Fine-tuning	Test	Bigrams	GPT2
Rally	Interview-I	596.03	25.00
	Interview-II	590.67	20.45
	Interview-III	601.66	21.52
	Interview-IV	628.43	25.45
	Interview-V	549.86	21.48
	Rally-I	575.04	19.82
	Rally-II	612.24	19.78
	Rally-III	561.30	19.53
	Rally-IV	603.62	19.82
	Rally-V	557.89	18.23
Interview	Interview-I	433.75	22.67
	Interview-II	450.34	20.14
	Interview-III	400.06	22.95
	Interview-IV	463.66	23.44
	Interview-V	406.60	21.78
	Rally-I	523.08	24.41
	Rally-II	523.53	26.59
	Rally-III	498.21	23.26
	Rally-IV	524.65	25.27
	Rally-V	494.36	22.60

- https://www.rev.com/blog/transcripts/i-have-been-to-the-mountaintop-speech-transcript-martin-luther-king-jr

- https://www.rev.com/blog/transcripts/the-other-america-speech-transcript-martin-luther-king-jr

- https://www.rev.com/blog/transcripts/the-american-dream-july-4th-speech-transcript-martin-luther-king-jr

- https://www.smu.edu/News/2014/mlk-at-smu-transcript-17march1966

- https://www.crmvet.org/docs/otheram.htm

URLs of transcripts for Nelson Mandela:

- https://blogs.lse.ac.uk/africaatlse/2013/12/06/full-text-of-nelson-mandela-speech-at-lse-on-6-april-2000/

- https://www.news24.com/news24/Columnists/GuestColumn/nelson-mandelas-speech-on-11-february-1990-i-stand-here-before-you-as-a-humble-servant-20200210

- https://www.sbs.com.au/news/transcript-nelson-mandela-speech-i-am-prepared-to-die

- http://db.nelsonmandela.org/speeches/pub_view.asp?pg=item&ItemID=NMS1522

- https://www.weforum.org/agenda/2013/12/nelson-mandelas-address-to-davos-1992/

URLs of transcripts for Barack Obama:

- https://www.rev.com/blog/transcripts/barack-obama-farewell-address-transcript-classic-speech-transcripts

- https://www.rev.com/blog/transcripts/barack-obama-campaign-speech-for-joe-biden-transcript-orlando-october-27

- https://www.rev.com/blog/transcripts/barack-obama-campaign-speech-for-joe-biden-transcript-miami-fl-november-2

- https://www.rev.com/blog/transcripts/barack-obama-florida-rally-speech-transcript-for-joe-biden-october-24

- https://www.rev.com/blog/transcripts/barack-obama-campaign-rally-for-joe-biden-kamala-harris-speech-transcript-october-21

URLs of transcripts for Bernie Sanders:

- https://www.rev.com/blog/transcripts/bernie-sanders-ann-arbor-campaign-speech-for-joe-biden-kamala-harris-october-5

- https://www.rev.com/blog/transcripts/bernie-sanders-nh-rally-speech-for-joe-biden-kamala-harris-october-3

- https://www.rev.com/blog/transcripts/bernie-sanders-speech-transcript-trumps-threat-to-our-democracy

- https://www.rev.com/blog/transcripts/bernie-sanders-st-louis-rally-speech-transcript-march-9-2020

- https://www.vox.com/2019/6/12/18663217/bernie-sanders-democratic-socialism-speech-transcript

URLs of transcripts for Donald Trump:

- https://www.rev.com/blog/transcripts/donald-trump-gastonia-nc-rally-speech-transcript-october-21

- https://www.rev.com/blog/transcripts/donald-trump-rally-transcript-tucson-arizona-october-19

- https://www.rev.com/blog/transcripts/donald-trump-macon-georgia-rally-speech-transcript-october-16

- https://www.rev.com/blog/transcripts/donald-trump-middletown-pa-rally-speech-transcript-sept-26-first-rally-after-scotus-nomination

- https://www.rev.com/blog/transcripts/donald-trump-newport-news-virginia-campaign-rally-transcript-september-25

A.3.5 Statistics describing data and detailed results for Experiment 2

Table A.11: Figures describing the transcripts employed in Experiment 2: time duration, number of tokens and number of unique tokens are reported for each such speech transcript.

Subject	Transcript	Duration	# Tokens	# Unique Tokens
Joe Biden	I	0 : 32 : 23	4,647	1,074
	II	0 : 41 : 39	5,446	1,140
	III	0 : 25 : 00	9,490	1,895
	IV	0 : 43 : 36	6,801	1,381
	V	0 : 34 : 05	5,211	1,226
Donald Trump	I	1 : 17 : 37	15,200	1,967
	II	0 : 56 : 17	10,501	1,614
	III	1 : 43 : 43	20,865	2,300
	IV	1 : 13 : 01	14,056	1,945
	V	1 : 18 : 19	14,806	1,896
Barack Obama	I	0 : 56 : 39	5,594	1,479
	II	0 : 38 : 15	6,298	1,252
	III	0 : 38 : 45	5,526	1,153
	IV	0 : 45 : 55	6,981	1,312
	V	0 : 36 : 07	5,390	1,159
Bernie Sanders	I	0 : 35 : 33	4,164	969
	II	0 : 29 : 51	3,785	849
	III	0 : 34 : 54	4,451	1,088
	IV	0 : 43 : 27	5,387	1,039
	V	0 : 44 : 46	4,501	1,286
Bill Gates	I	0 : 35 : 53	3,503	944
	II	0 : 17 : 20	1,679	577
	III	0 : 24 : 07	2,350	779
	IV	0 : 22 : 04	2,152	744
	V	0 : 30 : 07	2,896	1,018
Nelson Mandela	I	0 : 40 : 17	3,844	1,113
	II	0 : 29 : 45	1,740	617
	III	3 : 00 : 00	15,682	2,702
	IV	1 : 43 : 21	7,741	1,654
	V	0 : 40 : 16	3,020	963
Martin Luther King	I	0 : 42 : 51	5,197	1,102
	II	0 : 46 : 56	6,471	1,315
	III	0 : 43 : 48	6,287	1,456
	IV	0 : 40 : 38	8,256	1,697
	V	0 : 47 : 54	6,332	1,324
Boris Johnson	I	0 : 51 : 42	4,397	1,123
	II	0 : 20 : 35	2,758	764
	III	0 : 17 : 47	1,960	659
	IV	0 : 17 : 00	2,375	896
	V	0 : 38 : 22	4,530	1,273

Table A.12: Detailed perplexity scores for transcripts employed in Experiment 2.

Subject	Transcript	Bigrams	GPT2 Trial 1	GPT2 Trial 2
J. Biden	I	727.74	27.40	42.29
	II	653.82	27.31	43.59
	III	612.58	29.33	45.70
	IV	733.25	28.40	44.01
	V	751.25	27.87	46.28
D. Trump	I	575.04	19.82	42.58
	II	612.24	19.78	44.86
	III	561.30	19.64	41.37
	IV	603.62	19.78	43.29
	V	557.89	18.40	39.94
B. Obama	I	583.17	33.76	40.35
	II	496.75	15.11	36.10
	III	543.15	19.30	42.23
	IV	551.97	16.50	38.99
	V	519.50	15.77	38.33
B. Sanders	I	580.64	17.17	25.97
	II	628.01	16.40	26.47
	III	616.07	29.61	35.69
	IV	650.36	22.51	31.64
	V	553.45	26.02	29.76
B. Gates	I	497.05	37.20	43.04
	II	553.34	27.35	38.74
	III	425.17	38.75	47.74
	IV	452.09	41.35	50.71
	V	385.54	29.50	36.32
N. Mandela	I	1048.74	44.77	44.40
	II	845.18	29.50	35.69
	III	570.43	40.28	44.18
	IV	812.68	36.05	38.45
	V	842.22	37.78	37.26
M. L. King	I	850.19	31.00	42.84
	II	595.78	16.71	39.54
	III	711.06	35.52	49.84
	IV	737.79	29.24	45.82
	V	643.43	17.95	37.67
B. Johnson	I	391.33	40.76	54.69
	II	418.57	37.87	53.89
	III	435.47	62.44	83.04
	IV	455.18	53.81	59.71
	V	509.61	58.05	68.19

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... others (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283).

Agirre, A. G., Laparra, E., Rigau, G., & Donostia, B. C. (2012). Multilingual central repository version 3.0: upgrading a very large lexical knowledge base. In *Gwc 2012 6th international global wordnet conference* (p. 118).

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., & Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of naacl* (pp. 19–27). Association for Computational Linguistics.

Agirre, E., & De Lacalle, O. L. (2003). Clustering wordnet word senses. In *Ranlp* (Vol. 260, pp. 121–130).

Agirre, E., de Lacalle, O. L., & Soroa, A. (2014, March). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, *40*(1), 57–84. Retrieved from https://aclanthology.org/J14-1003 doi: 10.1162/COLI_a_00164

Ahn, D. (2006). The stages of event extraction. In *Proceedings of the workshop on annotating and reasoning about time and events* (pp. 1–8).

Ajani, G., Boella, G., Lesmo, L., Martin, M., Mazzei, A., Radicioni, D. P., & Rossi, P. (2010). Multilevel legal ontologies. In *Semantic processing of legal texts* (pp. 136–154). Springer.

Ali Awan, M. D., Ali, S., Samad, A., Iqbal, N., Saad Missen, M. M., & Ullah, N. (2021). Sentence classification using n-grams in urdu language text. *Scientific Programming*, *2021*.

Ansell, A., Bravo-Marquez, F., & Pfahringer, B. (2019). An elmo-inspired approach

to semdeep-5's word-in-context task. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2019*, *10*(2), 62–66.

Aribandi, V., Tay, Y., Schuster, T., Rao, J., Zheng, H. S., Mehta, S. V., ... others (2021). Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2111.10952*.

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, *6*, 483–495.

Baddeley, A. D. (1966a). The influence of acoustic and semantic similarity on long-term memory for word sequences. *The Quarterly journal of experimental psychology*, *18*(4), 302–309.

Baddeley, A. D. (1966b). Short-term memory for word sequences as a function of acoustic, semantic and formal similarity. *Quarterly Journal of Experimental Psychology*, *18*(4), 362-365. doi: 10.1080/14640746608400055

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Becker, J. T., Boiler, F., Lopez, O. L., Saxton, J., & McGonigle, K. L. (1994). The natural history of Alzheimer's disease: description of study cohort and accuracy of diagnosis. *Archives of Neurology*, *51*(6), 585–594.

Bedi, G., Carrillo, F., Cecchi, G. A., Slezak, D. F., Sigman, M., Mota, N. B., ... Corcoran, C. M. (2015). Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia*, *1*(1), 1–7.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003, mar). A neural probabilistic language model. *J. Mach. Learn. Res.*, *3*(null), 1137–1155.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166.

Benz, D., Körner, C., Hotho, A., Stumme, G., & Strohmaier, M. (2011). One tag to bind them all: Measuring term abstractness in social metadata. In *Proceedings of eswc* (pp. 360–374).

Berger, A., & Lafferty, J. (1999). Information retrieval as statistical translation. In *Proceedings of the 22nd annual international acm sigir conference on research and*

development in information retrieval (p. 222–229). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/312624.312681 doi: 10.1145/312624.312681

Bojanowski, P., Grave, É., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., ... others (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation* (pp. 12–58).

Bond, F., & Paik, K. (2012). A survey of wordnets and their licenses. *Small*, *8*(4), 5.

Bouraoui, Z., Camacho-Collados, J., Espinosa-Anke, L., & Schockaert, S. (2019). Modelling semantic categories using conceptual neighborhood. *arXiv preprint arXiv:1912.01220*.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 632–642).

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Budanitsky, A., & Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguists*, *32*(1), 13–47.

Buitelaar, P. (2000). Reducing lexical semantic complexity with systematic polysemous classes and underspecification. In *Proceedings of the 2000 naacl-anlp workshop on syntactic and semantic complexity in natural language processing systems-volume 1* (pp. 14–19).

Cabana, Á., Valle-Lisboa, J. C., Elvevåg, B., & Mizraji, E. (2011). Detecting order–disorder transitions in discourse: Implications for schizophrenia. *Schizophrenia Research*, *131*(1-3), 157–164.

Calabrese, A., Bevilacqua, M., & Navigli, R. (2020, July). Fatality killed the cat or: BabelPic, a multimodal dataset for non-concrete concepts. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4680–4686). Online: Association for Computational Linguistics. Retrieved from

https://www.aclweb.org/anthology/2020.acl-main.425

Camacho-Collados, J., & Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, *63*, 743–788.

Camacho-Collados, J., Pilehvar, M. T., Collier, N., & Navigli, R. (2017). Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 15–26).

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2015a). A framework for the construction of monolingual and cross-lingual word similarity datasets. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)* (Vol. 2, pp. 1–7).

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2015b). NASARI: a novel approach to a semantically-aware representation of items. In *Proceedings of naacl* (pp. 567–577).

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, *240*, 36–64.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 1–14).

Chen, J., & Palmer, M. S. (2009). Improving english verb sense disambiguation performance with linguistically motivated features and clear sense distinction boundaries. *Language Resources and Evaluation*, *43*(2), 181–208.

Chen, T., Xu, R., He, Y., & Wang, X. (2015). Improving distributed representation of word sense via wordnet gloss composition and context clustering. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)* (pp. 15–20).

Chen, X., Liu, Z., & Sun, M. (2014, 01). A unified model for word sense rep-

resentation and disambiguation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (p. 1025-1035). doi: 10.3115/v1/D14-1110

Chi, T.-C., & Chen, Y.-N. (2018). Cluse: Cross-lingual unsupervised sense embeddings. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 271–281).

Chi, T.-C., Shih, C.-Y., & Chen, Y.-N. (2018). Bcws: Bilingual contextual word similarity. *arXiv preprint arXiv:1810.08951*.

Chollet, F., et al. (2015). *Keras.*

Ciaramita, M., & Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 594–602).

Cohn, T. (2003). Performance metrics for word sense disambiguation. In *Proceedings of the australasian language technology workshop 2003* (pp. 86–93).

Çokal, D., Sevilla, G., Jones, W. S., Zimmerer, V., Deamer, F., Douglas, M., … others (2018). The language profile of formal thought disorder. *npj Schizophrenia*, *4*(1), 1–8.

Colla, D., Caselli, T., Basile, V., Mitrović, J., & Granitzer, M. (2020). Grupato at semeval-2020 task 12: Retraining mbert on social media and fine-tuned offensive language models. In *Proceedings of the fourteenth workshop on semantic evaluation* (pp. 1546–1554).

Colla, D., Mensa, E., Porporato, A., & Radicioni, D. P. (2018). Conceptual abstractness: from nouns to verbs. In *5th italian conference on computational linguistics, clic-it 2018* (pp. 70–75).

Colla, D., Mensa, E., & Radicioni, D. P. (2020a). Lesslex: Linking multilingual embeddings to sense representations of lexical items. *Computational Linguistics*, *46*(2), 289–333.

Colla, D., Mensa, E., & Radicioni, D. P. (2020b). Novel metrics for computing semantic similarity with sense embeddings. *Knowledge-Based Systems*, *206*, 106346.

Colla, D., Mensa, E., & Radicioni, D. P. (2020c). Sense Identification Dataset – SID.

Data in Brief, .

Collobert, R., & Weston, J. (2007). Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 560–567).

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

Copestake, A., & Briscoe, T. (1995). Semi-productive polysemy and sense extension. *Journal of semantics*, *12*(1), 15–67.

Covington, M. A., He, C., Brown, C., Naçi, L., McClain, J. T., Fjordbak, B. S., ... Brown, J. (2005). Schizophrenia and the structure of language: the linguist's view. *Schizophrenia research*, *77*(1), 85–98.

Davies, M. (2009). The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International journal of corpus linguistics*, *14*(2), 159–190.

de Boer, J. N., Brederoo, S. G., Voppel, A. E., & Sommer, I. E. (2020). Anomalies in language as a biomarker for schizophrenia. *Current opinion in psychiatry*, *33*(3), 212–218.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (pp. 248–255).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Di Fabio, A., Conia, S., & Navigli, R. (2019). Verbatlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 627–637).

Docherty, N. M., DeRosa, M., & Andreasen, N. C. (1996). Communication disturbances in schizophrenia and mania. *Archives of General Psychiatry*, *53*(4), 358–364.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., ... others (2021). Glam: Efficient scaling of language models with mixture-of-experts. *arXiv preprint arXiv:2112.06905*.

Edmonds, P., & Cotton, S. (2001). Senseval-2: overview. In *Proceedings of senseval-2 second international workshop on evaluating word sense disambiguation systems* (pp. 1–5).

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Elvevåg, B., Foltz, P. W., Weinberger, D. R., & Goldberg, T. E. (2007). Quantifying incoherence in speech: an automated methodology and novel application to schizophrenia. *Schizophrenia research*, *93*(1-3), 304–316.

Erk, K., McCarthy, D., & Gaylord, N. (2013). Measuring word meaning in context. *Computational Linguistics*, *39*(3), 511–554.

Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., & Fujita, H. (2020). Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*, *514*, 88–105.

Faruqui, M., & Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th conference of the european chapter of the association for computational linguistics* (pp. 462–471).

Faruqui, M., & Dyer, C. (2015). Non-distributional word vector representations. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)* (pp. 464–469).

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, *20*(1), 116–131.

Firth, J. R. (1935). The technique of semantics. *Transactions of the philological society*, *34*(1), 36–73.

Flekova, L., & Gurevych, I. (2016). Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of*

the 54th annual meeting of the association for computational linguistics (volume 1: Long papers) (pp. 2029–2041).

Frankenberg, C., Weiner, J., Schultz, T., Knebel, M., Degen, C., Wahl, H.-W., & Schroeder, J. (2019). Perplexity –a new predictor of cognitive changes in spoken language?– results of the Interdisciplinary Longitudinal Study on Adult Development and Aging (ILSE). *Linguistics Vanguard*, *5*(2), 1–10.

Fritsch, J., Wankerl, S., & Nöth, E. (2019). Automatic diagnosis of Alzheimer's disease using neural network language models. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 5841–5845).

Gale, W. A., & Church, K. W. (1994). What's wrong with adding one. *Corpus-based research into language: In honour of Jan Aarts*, 189–200.

Gentner, D., & Smith, L. (2012). Analogical reasoning. *Encyclopedia of human behavior*, *130*, 130.

Gerz, D., Vulić, I., Hill, F., Reichart, R., & Korhonen, A. (2016). Simverb-3500: A large-scale evaluation set of verb similarity. In J. Su, X. Carreras, & K. Duh (Eds.), *Proceedings of the 2016 conference on empirical methods in natural language processing (emnlp)* (p. 2173-2182). The Association for Computational Linguistics.

Goikoetxea, J., Soroa, A., & Agirre, E. (2018, June). Bilingual embeddings with random walks over multilingual wordnets. *Knowledge-Based Systems*, *150*(C), 218–230. doi: 10.1016/j.knosys.2018.03.017

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, *10*(1), 1–309.

Goldstone, R. L. (1994). The role of similarity in categorization: Providing a groundwork. *Cognition*, *52*(2), 125–157.

Goodglass, H., & Kaplan, E. (1983). *Boston diagnostic aphasia examination booklet*. Lea & Febiger.

Harman, D. (1993). Overview of the first trec conference. In *Proceedings of the 16th annual international acm sigir conference on research and development in information retrieval* (pp. 36–47).

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Havasi, C., Speer, R., & Alonso, J. (2007). ConceptNet: A lexical resource for common sense knowledge. *Recent advances in natural language processing V: selected papers from RANLP*, *309*, 269.

Hiemstra, D. (2001). *Using language models for information retrieval*. Citeseer.

Hill, F., Korhonen, A., & Bentz, C. (2014). A quantitative empirical analysis of the abstract/concrete distinction. *Cognitive Science*, *38*(1), 162–177.

Hill, F., Reichart, R., & Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), 665–695.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *28th annual meeting of the association for computational linguistics* (pp. 268–275).

Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E. G., & Milios, E. (2006). Information retrieval by semantic similarity. *International journal on semantic Web and information systems (IJSWIS)*, *2*(3), 55–73.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hu, R., Li, S., & Liang, S. (2019). Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th conference of the association for computational linguistics* (pp. 3899–3908).

Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1* (pp. 873–882).

Huang, L., Sun, C., Qiu, X., & Huang, X. (2019). Glossbert: Bert for word sense disambiguation with gloss knowledge. *arXiv preprint arXiv:1908.07245*.

Huff, F. J., Becker, J., Belle, S., Nebes, R., Holland, A., & Boller, F. (1987). Cognitive deficits and clinical diagnosis of Alzheimer's disease. *Neurology*, *37*(7), 1119–1119.

Iacobacci, I., & Navigli, R. (2019). Lstmembed: Learning word and sense representations from a large semantically annotated corpus with long short-term memories. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1685–1695).

Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2015). Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (Vol. 1, pp. 95–105).

Ivanov, B., Musa, M., & Dulamragchaa, U. (2021). Mongolian spelling error correction using word ngram method. In *Advances in intelligent information hiding and multimedia signal processing* (pp. 94–101). Springer.

Jawahar, G., Sagot, B., & Seddah, D. (2019). What does bert learn about the structure of language? In *Acl 2019-57th annual meeting of the association for computational linguistics.*

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Jimenez, S., Becerra, C., Gelbukh, A., Bátiz, A. J. D., & Mendizábal, A. (2013). Softcardinality-core: Improving text overlap with distributional measures for semantic textual similarity. In *Proceedings of *sem 2013* (Vol. 1, pp. 194–201).

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing. vol. 3*. Prentice Hall.

Kaisser, M., & Webber, B. (2007). Question answering based on semantic roles. In *Acl 2007 workshop on deep linguistic processing* (pp. 41–48).

Kneser, R., & Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing* (Vol. 1, pp. 181–184).

Kwong, O. O. (2008). A preliminary study on the impact of lexical concreteness on word sense disambiguation. In *Proceedings of the 22nd pacific asia conference on language, information and computation* (pp. 235–244).

Lamperti, G., & Zanella, M. (2006). Flexible diagnosis of discrete-event systems by similarity-based reasoning techniques. *Artificial Intelligence*, *170*(3), 232–297. doi: https://doi.org/10.1016/j.artint.2005.08.002

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2-3), 259–284.

Larson, M. K., Walker, E. F., & Compton, M. T. (2010). Early signs, diagnosis and therapeutics of the prodromal phase of schizophrenia and related psychotic disorders. *Expert review of neurotherapeutics*, *10*(8), 1347–1359.

Lavie, A., & Denkowski, M. J. (2009). The meteor metric for automatic evaluation of machine translation. *Machine translation*, *23*(2-3), 105–115.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on systems documentation* (p. 24–26). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/318723.318728 doi: 10.1145/318723.318728

Leturia, I. (2012). Evaluating different methods for automatically collecting large general corpora for basque from the web. In *Proceedings of coling 2012* (pp. 1553–1570).

Leviant, I., & Reichart, R. (2015a). Judgment language matters: Multilingual vector space models for judgment language aware lexical semantics. *CoRR, abs/1508.00106*.

Leviant, I., & Reichart, R. (2015b). Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv preprint arXiv:1508.00106*.

Levine, Y., Lenz, B., Dagan, O., Ram, O., Padnos, D., Sharir, O., ... Shoham,

Y. (2019). Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*.

Li, P., Mao, K., Yang, X., & Li, Q. (2019). Improving relation extraction with knowledge-attention. *arXiv preprint arXiv:1910.02724*.

Li, W., & McCallum, A. (2005). Semi-supervised sequence modeling with syntactic topic models. In *Aaai* (Vol. 5, pp. 813–818).

Lieto, A., Mensa, E., & Radicioni, D. P. (2016a). A Resource-Driven Approach for Anchoring Linguistic Resources to Conceptual Spaces. In *Xvth international conference of the italian association for artificial intelligence, genova, italy, november 29 – december 1, 2016, proceedings* (Vol. 10037, pp. 435–449). Springer. doi: 10.1007/978-3-319-49130-1

Lieto, A., Mensa, E., & Radicioni, D. P. (2016b). Taming sense sparsity: a common-sense approach. In *Proceedings of third italian conference on computational linguistics (clic-it 2016) & fifth evaluation campaign of natural language processing and speech tools for italian. final workshop (EVALITA 2016), napoli, italy, december 5-7, 2016.* (pp. 435–449).

Lieto, A., Radicioni, D. P., & Rho, V. (2017). Dual PECCS: A Cognitive System for Conceptual Representation and Categorization. *Journal of Experimental & Theoretical Artificial Intelligence*, *29*(2), 433-452. Retrieved from http://dx.doi.org/10.1080/0952813X.2016.1198934 doi: 10.1080/0952813X.2016.1198934

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. In *2015 ieee 14th international conference on cognitive informatics & cognitive computing (icci* cc)* (pp. 136–140).

Lison, P., & Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Liu, H. (2017). Sentiment analysis of citations using word2vec. *arXiv preprint arXiv:1704.00177*.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating Wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lopez, O. L., Swihart, A., Becker, J. T., Reinmuth, O., Reynolds, C., Rezek, D., & Daly, F. (1990). Reliability of NINCDS-ADRDA clinical criteria for the diagnosis of Alzheimer's disease. *Neurology*, *40*(10), 1517–1517.

Loureiro, D., & Jorge, A. (2019a, July). Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5682–5691). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1569 doi: 10.18653/v1/P19-1569

Loureiro, D., & Jorge, A. (2019b). Liaad at semdeep-5 challenge: Word-in-context (wic). In *Proceedings of the 5th workshop on semantic deep learning (semdeep-5)* (pp. 1–5).

Loureiro, D., Jorge, A. M., & Camacho-Collados, J. (2022). Lmms reloaded: Transformer-based sense embeddings for disambiguation and beyond. *Artificial Intelligence*, 103661.

MacWhinney, B. (2014). *The CHILDES project: Tools for analyzing talk, Volume II: The database*. Psychology Press.

MacWhinney, B. (2017). Tools for analyzing talk part 1: The CHAT transcription format. *Carnegie*, 1–115.

Mancini, M., Camacho-Collados, J., Iacobacci, I., & Navigli, R. (2017). Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of the 21st conference on computational natural language learning (conll 2017)* (pp. 100–111).

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Manning, C. D. (2015). Computational linguistics and deep learning. *Computational Linguistics*, *41*(4), 701–707.

Marconi, D. (1997). *Lexical competence*. MIT Press.

Marshall, M., Lewis, S., Lockwood, A., Drake, R., Jones, P., & Croudace, T. (2005).

Association between duration of untreated psychosis and outcome in cohorts of first-episode patients: a systematic review. *Archives of general psychiatry*, *62*(9), 975–983.

Maru, M., Scozzafava, F., Martelli, F., & Navigli, R. (2019, November). SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3534–3540). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1359 doi: 10.18653/v1/D19-1359

McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. In *Advances in neural information processing systems* (pp. 6294–6305).

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., ... others (2012). Interchanging lexical resources on the Semantic Web. *Language Resources and Evaluation*, *46*(4), 701–719.

McCrae, J. P., Rademaker, A., Rudnicka, E., & Bond, F. (2020). English wordnet 2020: Improving and extending a wordnet for english using an open-source methodology. In *proceedings of the lrec 2020 workshop on multimodal wordnets (mmw2020)* (pp. 14–19).

McCrae, J. P., Wood, I., & Hicks, A. (2017). The colloquial wordnet: Extending princeton wordnet with neologisms. In *International conference on language, data and knowledge* (pp. 194–202).

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th signll conference on computational natural language learning* (pp. 51–61).

Mensa, E., Porporato, A., & Radicioni, D. P. (2018). Annotating concept abstractness by common-sense knowledge. In C. Ghidini, B. Magnini, A. Passerini, & P. Traverso (Eds.), *Ai*ia 2018 – advances in artificial intelligence* (pp. 415–428). Cham: Springer International Publishing.

Mensa, E., Radicioni, D. P., & Lieto, A. (2017, August). Merali at semeval-2017 task 2 subtask 1: a cognitively inspired approach. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 236–240). Vancouver, Canada: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/S17-2038

Mihalcea, R., & Moldovan, D. I. (2001). Automatic generation of a coarse grained wordnet. In *Proceedings of the siglex workshop on "wordnet and other lexical resources: Applications, extensions and customizations" held in conjunction with naacl*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mikolov, T., Yih, W.-t., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 746–751). Atlanta, Georgia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N13-1090

Miller, D. R. H., Leek, T., & Schwartz, R. M. (1999). A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval* (p. 214–221). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/312624.312680 doi: 10.1145/312624.312680

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, *6*(1), 1–28.

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., & Thomas, R. G. (1994). Using a semantic concordance for sense identification. In *Human language*

technology: Proceedings of a workshop held at plainsboro, new jersey, march 8-11, 1994.

Mohammad, S. M., & Hirst, G. (2012). Distributional measures of semantic distance: A survey. *arXiv preprint arXiv:1203.1858*.

Moro, A., & Navigli, R. (2015). Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (semeval 2015)* (pp. 288–297).

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244.

Mota, N. B., Copelli, M., & Ribeiro, S. (2017). Thought disorder measured as random speech structure classifies negative symptoms and schizophrenia diagnosis 6 months in advance. *npj Schizophrenia*, *3*(1), 1–10.

Mota, N. B., Vasconcelos, N. A., Lemos, N., Pieretti, A. C., Kinouchi, O., Cecchi, G. A., ... Ribeiro, S. (2012). Speech graphs provide a quantitative measure of thought disorder in psychosis. *PloS One*, *7*(4), e34928.

Mrkšić, N., Séaghdha, D. Ó., Thomson, B., Gasic, M., Barahona, L. M. R., Su, P.-H., ... Young, S. (2016). Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 142–148).

Mu, J., Bhat, S., & Viswanath, P. (2017). Geometry of polysemy. In *5th international conference on learning representations, ICLR 2017, toulon, france, april 24-26, 2017, conference track proceedings*. Retrieved from https://openreview.net/forum?id=HJpfMIFll

Navigli, R. (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics* (pp. 105–112).

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, *41*(2), 10.

Navigli, R., Jurgens, D., & Vannella, D. (2013). Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second joint conference on lexical and com-*

putational semantics (sem), volume 2: Proceedings of the seventh international workshop on semantic evaluation (semeval 2013)* (pp. 222–231).

Navigli, R., & Martelli, F. (2019). An overview of word and sense similarity. *Natural Language Engineering*, *25*(6), 693–714.

Navigli, R., & Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 216–225).

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Neelakantan, A., Shankar, J., Passos, A., & McCallum, A. (2014a). Efficient non-parametric estimation of multiple embeddings per word in vector space. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Emnlp* (pp. 1059–1069). ACL.

Neelakantan, A., Shankar, J., Passos, A., & McCallum, A. (2014b). Efficient non-parametric estimation of multiple embeddings per word in vector space. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing(emnlp)* (pp. 1059–1069). ACL.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (2004). The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, *36*(3), 402–407.

Neuman, Y., Assaf, D., Cohen, Y., Last, M., Argamon, S., Howard, N., ... others (2013). Metaphor identification in large texts corpora. *PLOS ONE*, *8*(4), 1–9.

Ng, H. T., Wang, B., & Chan, Y. S. (2003). Exploiting parallel texts for word sense disambiguation: An empirical study. In *Proceedings of the 41st annual meeting on association for computational linguistics-volume 1* (pp. 455–462).

Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, memory, and cognition*, *14*(3), 510.

Pal, A. R., & Saha, D. (2014). An approach to automatic text summarization using wordnet. In *2014 ieee international advance computing conference (iacc)* (pp. 1169–1173).

Palmer, M., Babko-Malaya, O., & Dang, H. T. (2004). Different sense granularities

for different applications. In *Proceedings of the 2nd international workshop on scalable natural language understanding (scanalu 2004) at hlt-naacl 2004* (pp. 49–56).

Patwardhan, S., & Pedersen, T. (2006). Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the workshop on making sense of sense: Bringing psycholinguistics and computational linguistics together.*

Pedersen, T., Banerjee, S., & Patwardhan, S. (2005). Maximizing semantic relatedness to perform word sense disambiguation. *University of Minnesota supercomputing institute research report UMSI*, *25*, 2005.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pelevina, M., Arefiev, N., Biemann, C., & Panchenko, A. (2016). Making sense of word embeddings. In *Proceedings of the 1st workshop on representation learning for nlp* (pp. 174–183).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing(emnlp)* (Vol. 14, pp. 1532–1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of naacl-hlt* (pp. 2227–2237).

Pianta, E., Bentivogli, L., & Girardi, C. (2002). Multiwordnet: developing an aligned multilingual database. In *First international conference on global wordnet* (pp. 293–302).

Pilehvar, M. T., & Camacho-Collados, J. (2019). WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 1267–1273).

Pilehvar, M. T., & Collier, N. (2016). De-Conflated Semantic Representations. In *Proceedings of the 2016 conference on empirical methods in natural language*

processing (emnlp) (pp. 1680–1690).

Pilehvar, M. T., & Navigli, R. (2015). From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, *228*, 95-128.

Pradhan, S., Loper, E., Dligach, D., & Palmer, M. (2007). Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the fourth international workshop on semantic evaluations (semeval-2007)* (pp. 87–92).

Pustejovsky, J. (1991). The generative lexicon. *Computational linguistics*, *17*(4), 409–441.

Qu, C., Yang, L., Qiu, M., Croft, W. B., Zhang, Y., & Iyyer, M. (2019). Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (pp. 1133–1136).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 1, long papers* (pp. 99–110).

Reisinger, J., & Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 109–117).

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In C. R. Perrault (Ed.), *Proceedings of the 14th ijcai* (pp. 448–453). Montréal (Canada).

Resnik, P., & Yarowsky, D. (1999). Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural language engineering*, *5*(2), 113–133.

Roventini, A., Alonge, A., Bertagna, F., Calzolari, N., Zampolli, A., Girardi, C., ... Cancila, J. (1998). " italwordnet": Building a large semantic database for the automatic treatment of italian. *" ItalWordNet"*, 1000–1047.

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*(10), 627–633.

Rudnicka, E. K., Witkowski, W., & Kaliński, M. (2015). Towards the methodology for extending princeton wordnet. *Cognitive Studies*(15).

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Scarlini, B., Pasini, T., & Navigli, R. (2020a). SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proceedings of the thirty-fourth conference on artificial intelligence* (pp. 8758–8765). Association for the Advancement of Artificial Intelligence.

Scarlini, B., Pasini, T., & Navigli, R. (2020b). With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation. In *Proceedings of the 2020 conference on empirical methods in natural language processing*. Association for Computational Linguistics.

Schaeffer, B., & Wallace, R. (1969). Semantic similarity and the comparison of word meanings. *Journal of Experimental Psychology*, *82*(2), 343.

Schütze, H. (1998). Automatic word sense discrimination. *Computational linguistics*, *24*(1), 97–123.

Schütze, H., & Pedersen, J. O. (1997). A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing & Management*, *33*(3), 307–318.

Schwartz, H. A., & Gomez, F. (2011). Evaluating semantic metrics on tasks of concept similarity. In *Proceedings of the international florida artificial intelligence research society conference (flairs)* (pp. 299–304).

Scott, S., & Matwin, S. (1998). Text classification using wordnet hypernyms. In *Usage of wordnet in natural language processing systems.*

Scozzafava, F., Raganato, A., Moro, A., & Navigli, R. (2015). Automatic identification and disambiguation of concepts and named entities in the multilingual wikipedia. In *Congress of the italian association for artificial intelligence* (pp.

357–366).

Shao, Y. (2017). HCTI at SemEval-2017 Task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 130–133).

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. In *Proceedings of the 20th nordic conference of computational linguistics (nodalida 2015)* (pp. 239–243).

Singh, M., Jakhar, A. K., & Pandey, S. (2021). Sentiment analysis on the impact of coronavirus in social life using the bert model. *Social Network Analysis and Mining*, *11*(1), 1–11.

Snyder, B., & Palmer, M. (2004). The english all-words task. In *Proceedings of senseval-3, the third international workshop on the evaluation of systems for the semantic analysis of text* (pp. 41–43).

Soler, A. G., Apidianaki, M., & Allauzen, A. (2019). LIMSI-MULTISEM at the IJCAI SemDeep-5 WiC Challenge: Context Representations for Word Usage Similarity Estimation. In *Proceedings of the 5th workshop on semantic deep learning (semdeep-5)* (pp. 6–11).

Souza, F., Nogueira, R., & Lotufo, R. (2019). Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Speer, R., & Chin, J. (2016). *An ensemble method to produce high-quality word embeddings.*

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Aaai* (pp. 4444–4451).

Speer, R., & Lowry-Duda, J. (2017). Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 85–89). Association for Computational Linguistics. Retrieved from http://aclweb.org/anthology/S17-2008 doi: 10.18653/v1/S17-2008

Stolcke, A., & Shriberg, E. (1996). Statistical language modeling for speech disfluencies. In *1996 ieee international conference on acoustics, speech, and signal processing conference proceedings* (Vol. 1, pp. 405–408).

Sun, R. (1995). Robust reasoning: integrating rule-based and similarity-based

reasoning. *Artificial Intelligence*, *75*(2), 241 – 295. doi: https://doi.org/10.1016/0004-3702(94)00028-Y

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tauchmann, C., & Mieskes, M. (2020). Language agnostic automatic summarization evaluation. In *Proceedings of the 12th language resources and evaluation conference* (pp. 6656–6662).

Tenney, I., Das, D., & Pavlick, E. (2019, July). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4593–4601). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1452 doi: 10.18653/v1/P19-1452

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., ... Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Tomuro, N. (2001). Tree-cut and a lexicon based on systematic polysemy. In *Second meeting of the north american chapter of the association for computational linguistics*.

Tripodi, R., & Pelillo, M. (2017). A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, *43*(1), 31–70.

Turney, P. D., Neuman, Y., Assaf, D., & Cohen, Y. (2011). Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 680–690).

Tversky, A. (1977). Features of similarity. *Psychological review*, *84*(4), 327.

Tversky, A., & Kahneman, D. (1973). Availability: a heuristic for judging frequency and probability. *Cognitive Psychology*, *5*, 207–232.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *science*, *185*(4157), 1124–1131.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85.

Vu, T., & Parker, D. S. (2016). K-embeddings: Learning conceptual embeddings for words using context-embeddings: Learning conceptual embeddings for words using context. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1262–1267).

Walenski, M., Weickert, T. W., Maloof, C. J., & Ullman, M. T. (2010). Grammatical processing in schizophrenia: Evidence from morphology. *Neuropsychologia*, *48*(1), 262–269.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., … Bowman, S. R. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Webber, W., Moffat, A., & Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, *28*(4), 20.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., … Macherey, K. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xu, H., Liu, B., Shu, L., & Yu, P. S. (2019). Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*.

Yaghoobzadeh, Y., & Schütze, H. (2016). Intrinsic subspace evaluation of word embedding representations. *arXiv preprint arXiv:1606.07902*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Zhu, Z., Bernhard, D., & Gurevych, I. (2010). A monolingual tree-based transla-

tion model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 1353–1361).