

Translating Italian to LIS in the rail stations

Alessandro Mazzei

Dipartimento di Informatica, Università degli Studi di Torino
Via Pessinetto 12, 10149, Torino, Italy
mazzei@di.unito.it

Abstract

This paper presents an ongoing project about the symbolic translation from Italian to Italian Signed Language (LIS) in the rail stations domain. We describe some technical issues in the generation side of the translation, i.e. the use of XML templates for microplanning, the implementation of some LIS linguistic features in the grammar.

1 Introduction

Several commercial and research projects use avatars for automatic translation into signed languages (SLs) and most of these projects investigate on relatively small domains in which translation may perform quite well. Among them: post office announcements (Cox et al., 2002), weather forecasting (Verlinden et al., 2001; Mazzei et al., 2013), driver’s license renewal (San-Segundo et al., 2012), and train announcements (Segouat and Braffort, 2009; Ebling, 2013). However, SLs still pose many challenges related to the specific linguistic features (e.g. no function words and articles) as well as to the specific communication channels (e.g. the characteristic use of the space).

LIS4ALL is a project for the automatic translation from Italian to LIS in the Italian rail stations domain. The domain is completely specified by the Rete Ferroviaria Italiana (RFI), which produced the manual *MAS* (Manuale degli Annunci Sonori), that describes the details of each specific message (RFI, 2011). *MAS* specifies 39 classes: 13 for arriving trains, 15 for leaving trains, 11 for special situations (e.g. strikes). The classes have been designed by a group of linguists to produce concise Italian messages. Full relative clauses, coordination and complex structures are avoided. As a consequence, the Italian rail stations domain is a controlled plain language. Note that the vocal rail

station messages are produced in real-time by using text-to-speech, and the textual input message is produced by a proprietary closed source software that uses raw data extracted from a database. In the project we had access only to the textual messages but we do not have access to raw data. As a consequence, LIS4ALL concerns automatic translation with NLG rather than uniquely NLG.

An initial study of the domain (5014 messages form 24 hours messages produced at the Torino Porta Nuova Station) has showed that four classes account for $\sim 85\%$ of total messages: these are A1: *simple arrive*, P1: *simple leave*, A2: *arrive on a different rail*, A3: *delayed arrive*. In this paper we discuss a symbolic translator designed to account for these four classes of messages.¹

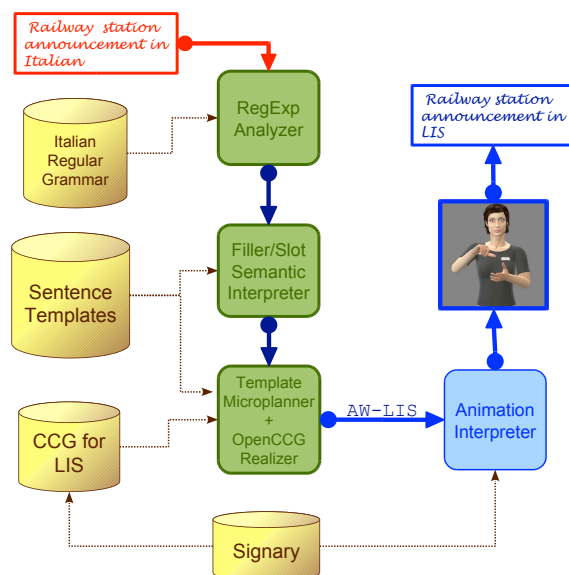


Figure 1: The architecture of the Italian to LIS translator.

Figure 1 illustrates the current architecture, thus a pipeline which includes five modules: (1) a regular expression parser for Italian; (2) a filler/slot

¹We plan to account for the remaining classes in the future.

based semantic interpreter; (3) a generator; (4) a spatial planner; (5) an avatar that performs the synthesis of the sequence of signs. On the basis of this architecture, our translator can be classified as semantic transfer based system (Hutchins and Somer, 1992). Indeed, since the source language is controlled, the translation is a deterministic process that poses a number of challenges just related to the target language, thus in the generation process.

Indeed, the LIS4ALL translator adopts the architecture designed in the ATLAS project (Mazzei et al., 2013) with two essential changes. The first change is related to the analysis of the Italian sentences: the domain corpus shows that typical railway station messages have several prepositional phrases which pose hard problems to conventional natural language parsers. As a consequence, we developed a domain specific regular expressions parser. The second change is related to the semantic representation. In ATLAS a logic semantics was adopted, that was the input of an expert system micro-planner. In contrast, in LIS4ALL we adopt a simpler non-recursive filler/slot semantics. For parsing, we have built four regular expressions corresponding to the four most frequent classes in the domain corpus. For each class, we designed a set of semantic slots that can be filled by domain lexical items (time, rail number, station names, etc.). So, the role of the semantic interpreter is just to extract the semantics of these Italian domain lexical items, and to convert them into a format that can be realized in LIS domain lexical items. The translation consists essentially in converting the filler/slot semantics produced by the semantic interpreter into a logic format that can be exploited in generation. In Table 1 we reported an Italian message, its translation into LIS, and the filler/slot semantics produced by the semantic interpreter.

In Section 2 we give some details about the transfer process to transform the filler/slot semantics in hybrid logic semantics; in Section 3 we describe the implementation in a Combinatory Categorical Grammar (CCG) of a number of specific LIS linguistic phenomena. Finally, Section 4 closes the paper with some considerations.

ITA: Il treno Regionale 10220 di Trenitalia delle ore 05:35 proveniente da Cuneo arrivera con un ritardo previsto di 10 minuti .

LIS: TRENO REGIONALE NUMERO 1 0 2 2 0 TRENITALIA POSS ORE 5 35 MATTINA CUNEO VENIRE , RITARDO 10 MINUTI PREVISTO ARRIVARE FUT_DEVE

```
sem: :type :A3
      :numero "10220"
      :impresa_ferroviaria "trainitaly"
      :categoria "REGIONAL"
      :località_di_provenienza "cuneo"
      :ampm "morning"
      :ora_arrivo "05:35"
      :hh "5"
      :mm "35"
      :tempo_ritardo "10"
```

Table 1: An ITA/LIS sentence of the class A3 and its filler/slot semantics. GLOSSES are used to denote LIS signs. The underlined texts correspond to variable lexical items. Rough translation: *The regional train 10220 of Trenitalia arriving at 05:35am from Cuneo, will arrive with an expected delay of 10 minutes.*

2 Microplanning with XML transformations

Previous work on the symbolic translation of SL in rail stations domains adopted “video templates” for the generation side (Segouat and Braffort, 2009; Ebling, 2013). In contrast, our generator is more complex and adopts the standard pipeline architecture of the NLG (Reiter and Dale, 2000). The generator is composed by two elements: a template based microplanner and the OpenCCG realizer.

Following (Foster and White, 2004), we implemented a transformation based microplanner that is able to exploit the filler/slot structure produced by the semantic interpreter. The main idea is to recursively rewrite the semantics elements by using a number predefined XML templates. Four templates are used at the first stage to specify the main structures of the sentences plans, while seven templates are used at the second stage to specify the specific structures of a number of specific linguistic constructions, e.g. the rail number.

For the implementation of the microplanner we exploited the bidirectional nature of OpenCCG by adopting a bottom-up approach to build the XML templates. For each MAS class we choose an Italian sentence belonging to the class and we pro-

duce, with the help of a bilingual signer, the LIS translation of the sentence. In Table 1 we report the translation of a sentence belonging to the class A3 (delayed arrive).

By starting from the Italian/LIS translation of the sentence, we have followed four steps:

1. We have implemented the fragment of the grammar necessary to realize/parse the LIS sentence, i.e. to account the linguistic phenomena contained in the sentence (see Section 3).
2. We have obtained the hybrid logic formula expressing the linguistic meaning of the sentence by parsing the sentence.
3. We have modified the XML file containing the hybrid logic formula by introducing a number of *holes*. Each hole, implemented as a XML attribute, corresponds to a LIS lexical item. For instance, in the XML fragment

```
<diamond mode="SYN-NOUN-RMOD">
  <nom name="n4:number"/>
  <prop id="delay-amount" name="10"/>
</diamond>
```

that is the linguistic meaning of the number 10 and that corresponds to the delay amount, we have introduced the hole “delay-amount”. In this way, the XML processor will be able to rewrite the exact delay for all the sentences of the class A3.

4. We have designed a number of XML transformations in order to convert the filler/slot semantics produced by the interpreter to the corresponding logical formula. A single filler/slot semantic element will substitute the XML fragment corresponding to a single hole in the final logical formula.

So, in total we have designed a total amount of 11 XML transformations to account for all the filler/slot semantic elements of the four MAS classes. Note that some of these transformations are recursive. This is the case, for instance, of the train code: LIS signers realize the code with a sequence of digits rather than with a single number, as in Italian (see Table 1).

3 A CCG for LIS in rail stations

We have designed a new CCG for LIS in the rail stations domain starting from the CCG for LIS devised in ATLAS project (Mazzei et al., 2013).

SLs do not have adpositions and articles, and use pronouns and conjunctions in very specific ways (Brentani, 2010). As a consequence, a very challenging topic is the grammatical design of the modifiers. In contrast to vocal languages, where the modification of a noun with another noun is usually marked by adpositions, in SLs the proximity in the sentence is the only possible indicator of the modification². Indeed, noun modifications occurs often in the the rail station domain: in the LIS sentence of Table 1 there are five noun \leftarrow noun modifications, which are used to indicate train code (TRENTO \leftarrow NUMERO), train company (TRENTO \leftarrow TRENITALIA), scheduled time (TRENTO \leftarrow ORE), delay amount (TRENTO \leftarrow RITARDO). Our CCG design uses type-change for promoting a standard noun category (N) into a noun modifier category ($N \setminus N$). However, this design increases the ambiguity of the grammar, since a noun could be the modifier of all previous nouns. In order to mitigate the grammar ambiguity, we have enriched the noun category with a *type-change count* feature. The idea is to allow the modification of a noun only if this noun has not been obtained with another type-change. Formally:

$$TC_1 : N_{tc_1} \rightarrow N_{tc_0}$$

$$TC_2 : N_{tc_1} \rightarrow N_{tc_0} \setminus N_{tc_0}$$

In this way, the noun NUMERO cannot (1) be modified by the noun RITARDO, and in the same derivation (2) modify the noun TRENTO. From another point of view, the introduction of the type-change count feature constrains the hybrid logic dependency structure to be flat.

Another well known problem related to modifiers is their realization order. Similar to vocal languages, SLs have strong pragmatic preferences for specific modifiers order. The symbolic and statistical nature of OpenCCG allows to manage this issue by using a probabilistic approach. Indeed, it is possible to associate probabilities over logically equivalent derivations by using a language model (White, 2005). In order to use this feature, we have built an “artificial” corpus of 50 LIS sentences by using the four most frequent MAS templates. By using this corpus, we have built a trigrams based model language that derives the most natural modifiers sequence.

²Spatial agreement is another indication of syntactic agreement (Wright, 2008; Mazzei et al., 2013), but we did not yet model this feature in the actual CCG for LIS.

Another grammar issue concerns the lexical semantics. OpenCCG organizes the lexical items in an ontological structure. In the implementation of the LIS grammar we have used the backbone of the DOLCE ontology (Masolo et al., 2002), i.e. the LIS lexical items (~ 120 signs) have been classified under the top level categories of DOLCE. For instance, the semantic category *rail* has been collocated as a child of the DOLCE category *non agentive physical object*.

Another specific feature of the LIS CCG lexicon concerns the lexicalization of some station names. Previous approaches to SL translation in the rail station domain propose to fingerspell the names of the secondary stations (Segouat and Braffort, 2009; Ebling, 2013), i.e. the station which do not have a well known name in the national Deaf community. In contrast, we propose to exploit the virtual nature of the avatar by producing a classifier sign that generate dynamically new lexical items. We distinguish two kinds of rail stations: (1) In the case of a well-known station, the avatar uses the sign adopted by the Deaf community; (2) In contrast, in the case of less known station, the avatar realizes a classifier sign indicating a wide board while the name of the station will appear in written Italian “centered on the board” (Figure 2). Note that we had to modify the lexicalization mechanism of OpenCCG with a workaround in order to implement this feature. Indeed OpenCCG assumes a “closed lexicon”, i.e. assumes that the lexicon is a closed set completely specified in the grammar. We have introduced a post-processing lexical substitution procedure that replaces a generic sign for less known stations with the board sign, modified in real time with the name of the station. More details on the linguistic impact of the board sign are reported in (Geraci and Mazzei, 2013).

4 Summary and future work

We have described some issues related to the generation module of the symbolic translator from Italian to LIS designed in the LIS4ALL project. The main contribution of this paper is to show that the combination of a filler/slot semantics with a XML transformation-based microplanner is adequate to generate controlled domain languages.

A prototype of the translator has been implemented in *Clojure*³, that is a functional program-

³<http://clojure.org>



Figure 2: The sign for *Rebaudengo Fossata*, a less known station in Turin.

ming language that works on the Java virtual machine. Clojure exploits the the widespread use of Java by allowing (1) to call efficiently external Java libraries, and (2) to deploy software on different machines. Indeed, in order to implement the template based microplanner, we have used the *Enlive* library⁴, i.e. a selector based system primarily designed for web templating. Moreover, the OpenCCG realizer has been natively called from the Clojure code.

In the next future we plan to introduce in the generator the linguistic management of signing space since previous work have showed that CCG can compactly model this linguist feature (Wright, 2008; Mazzei et al., 2013).

Finally, we intend to evaluate the quality of our translator by using both task-based human evaluation (Mazzei et al., 2013) as well as metric-based automatic evaluation (Battaglino et al., 2015).

Acknowledgments

This work has been partially supported by the project LIS4ALL, partially funded by Regione Piemonte, Innovation Hub for ICT, 2011-2014, POR-FESR 07-13.

This work is dedicated to Leonardo Lesmo who substantially contributed to its realization.

References

Cristina Battaglino, Carlo Geraci, Vincenzo Lombardo, and Alessandro Mazzei. 2015. Prototyping and preliminary evaluation of sign language translation system in the railway domain. In *Universal Access in*

⁴<https://github.com/cgrand/enlive>

- Human-Computer Interaction. Access to Interaction - 9th International Conference, UAHCI 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part II*, pages 339–350.
- Dana Brentani, editor. 2010. *Sign Languages*. Cambridge University Press.
- Stephen Cox, Michael Lincoln, Judy Tryggvason, Melanie Nakisa, Mark Wells, Marcus Tutt, and Sanja Abbott. 2002. Tessa, a system to aid communication with deaf people. In *Proceedings of the fifth international ACM conference on Assistive technologies*, pages 205–212. ACM.
- Sara Ebling. 2013. Evaluating a swiss german sign language avatar among the deaf community. In *Third International Symposium on Sign Language Translation and Avatar Technology*, October.
- Mary Ellen Foster and Michael White. 2004. Techniques for text planning with XSLT. In *Proceedings of the Workshop on NLP and XML (NLPXML-2004): RDF/RDFS and OWL in Language Technology*, pages 1–8. Association for Computational Linguistics.
- Carlo Geraci and Alessandro Mazzei. 2013. Last train to “rebaudengo fossano”: The case of some names in avatar translation. In *Proceedings of the 6th Workshop on the Representation and Processing of Sign Languages*, pages 63–66.
- W. John Hutchins and Harold L. Somer. 1992. *An Introduction to Machine Translation*. London: Academic Press.
- Carlo Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, and Luc Schneider. 2002. Wonder Web deliverable D17. The Wonder Web Library of Foundational Ontologies and the DOLCE Ontology. Technical Report D17, ISTC-CNR.
- Alessandro Mazzei, Leonardo Lesmo, Cristina Battaglino, Mara Vendrame, and Monica Bucciarrelli. 2013. Deep natural language processing for italian sign language translation. In *AI*IA 2013: Advances in Artificial Intelligence - XIIIth International Conference of the Italian Association for Artificial Intelligence, Turin, Italy, December 4-6, 2013. Proceedings*, pages 193–204.
- E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.
- Rete Ferroviaria Italiana RFI. 2011. *Manuale Annunci Sonori - MAS* (<http://www.rfi.it/cms-file/allegati/rfi/MAS.pdf>). Rete Ferroviaria Italiana (RFI).
- Rubén San-Segundo, Juan Manuel Montero, R Córdoba, Valentin Sama, F Fernández, LF D’Haro, Verónica López-Ludeña, D Sánchez, and A García. 2012. Design, development and field evaluation of a spanish into sign language translation system. *Pattern Analysis and Applications*, 15(2):203–224.
- Jérémie Segouat and Annelies Braffort. 2009. Toward modeling sign language coarticulation. In *Gesture in Embodied Communication and Human-Computer Interaction, 8th International Gesture Workshop, GW 2009, Bielefeld, Germany, February 25-27, 2009, Revised Selected Papers*, pages 325–336.
- Margriet Verlinden, Corrie Tijsseling, and Han Frowein. 2001. A signing avatar on the WWW. In *Gesture and Sign Languages in Human-Computer Interaction, International Gesture Workshop, GW 2001, London, UK, April 18-20, 2001, Revised Papers*, pages 169–172.
- Michael White. 2005. Designing an Extensible API for Integrating Language Modeling and Realization. In *Proceedings of the Workshop on Software, Software ’05*, pages 47–64, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tony Wright. 2008. A combinatory categorial grammar of a fragment of american sign language. In *Proc. of the Texas Linguistics Society X Conference*. CSLI Publications.