Sign Language Generation with Expert Systems and CCG

Alessandro Mazzei
Dipartimento di Informatica
Università degli Studi di Torino
Corso Svizzera 185, 10185 Torino Italy
mazzei@di.unito.it

Abstract

This paper concerns the architecture of a generator for Italian Sign Language. In particular we describe a microplanner based on an expert-system and a combinatory categorial grammar used in realization.

1 Introduction

In this paper we present the main features of the generator used into a translation architecture from Italian to Italian Sign Language (Lingua Italiana dei Segni, henceforth LIS), that is the sign language used by the Italian deaf (signing) community (Volterra, 2004). Our generator consists of two modules: (i) SentenceDesigner, that is a rule-based microplanner; (ii) OpenCCG, that is a chart realizer (White, 2006). There are two main issues in this work. The first issue concerns the use of an expert system for microplanning. Most of our knowledge about LIS linguistics derives from discussions with linguists: expert systems allow for sharp modularization of this human knowledge. Moreover, expert-system allow us for easily updateable knowledge organization in cases of conflict or contradiction. The second issue in our work concerns the design of a combinatory categorial grammar (CCG) used by the realizer. This CCG accounts for a number of specific LIS phenomena as spatial verb-arguments agreement and NP coordination.[1]

[1]In this paper we present a *grammatical* account for spatial verb-arguments agreement. A different approach, that we are exploring too, is to consider space allocation as separate process that takes as input the syntactic structure, similar to prosody in vocal languages.

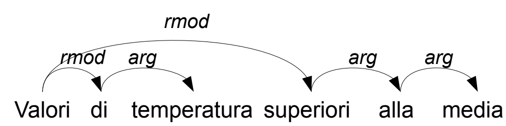

Figure 1: The (simplified) syntactic structure of the sentence "Valori di temperatura superiori alla media" (*Temperature values exceed the average*) produced by the TUP parser.

In order to reduce the difficulties of our project we concentrated on a specific application domain, i.e. weather forecasts: a group of linguists produced a small parallel corpus (300 sentences) of Italian-LIS sentences extracted from TV news and concerning weather forecasts. Building vocal-SL parallel corpora is a hard task: there are theoretical difficulties concerning the extra-video annotation. In particular, while there are standards for the representation of the phonological information of the signs, there are no standard ways to represent their morpho-syntactic inflections. The corpus has been used primarily to produce an electronic dictionary for the virtual interpreter consisting of about 1500 signs, that provides a lexicon for the realizer too. In contrast, most of the knowledge about LIS syntax comes from discussions with some linguists.

2 Parsing and Interpretation

Our interlingua translation system is a chain composed of four distinct modules, that are: (1) a dependency parser for Italian; (2) an ontology based semantic interpreter; (3) a generator; (4) a virtual actor that performs the synthesis of the final LIS sentence. In this Section we give some details about the

INLG 2012 Proceedings of the 7th International Natural Language Generation Conference, pages 105–109,
Utica, May 2012. ©2012 Association for Computational Linguistics

parser and the semantic interpreter, in Sections 3 and 4 we describe the generator.

In the first step, the syntactic structure of the source language is produced by the TUP, a rule-based parser (Lesmo, 2007). The TUP is based on a morphological dictionary of Italian (about $25,000$ lemmata) and a rule-based grammar, and it produces a *dependency tree*, that makes clear the structural syntactic relationships occurring between the words of the sentence. Each word in the source sentence is associated with a node of the tree, and the nodes are linked via labeled arcs that specify the syntactic role of the dependents with respect to their head (the parent node). In Figure 1 we show the syntactic analysis for the sentence "Valori di temperatura superiori alla media" (rough translation: *Temperature values exceed the average*). The edge label "ARG" indicates an *ARG*ument relation, i.e. an obligatory relation between the head and its argument. The edge label "RMOD" indicates a *R*estricting *MODifier* relation, i.e. a non obligatory relation from the head and its dependent (Bosco and Lombardo, 2004).

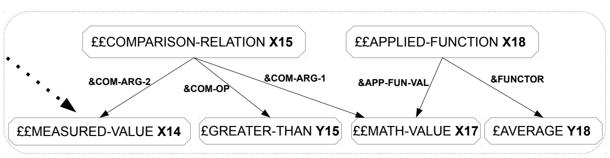

Figure 2: The fragment of the semantic network resulting from the interpretation of the sentence "Valori di temperatura superiori alla media".

The second step of the translation is the semantic interpretation: the syntax-semantics interface is based on ontologies (Lesmo et al., 2011). The knowledge in the ontology, which has been designed for this specific application, concerns the application domain, i.e. weather forecasts, as well as more general common knowledge about the world. Note that the ontology used by the semantic interpreter is not the same ontology used by the generator (microplanner and realizer): indeed, whilst the semantic interpreter ontology describes the linguistic knowledge of the Italian language, the generator describes the linguistic knowledge of the LIS. Starting from the lexical semantics of the words and on the basis of the dependency structure, a recursive function searches in the ontology providing a number of "connection paths" that represent the meaning. In fact, the final sentence meaning consists of a complex fragment of the ontology, i.e. a single connected semantic network (Lesmo et al., 2011). In Figure 2 we show a fragment of the semantic network resulting from the interpretation of the sentence "Valori di temperatura superiori alla media". The nodes of the network contain instances (prefix name £), concepts (prefix name ££) relations (prefix name &) from the ontology. In Figure 2 the nodes AVERAGE, GRATER-THAN are instances, the other nodes are concepts. Informally speaking, we can say that the semantic interpreter organizes the information of the semantic network as a number of "information chunks" that are weakly connected to the other parts of the network. In the network of Figure 2 we can distinguish two chunks. The paraphrase of these chuncks meanings is: there is a (temperature) value involved in a comparison (chunk 1) with a mathematical value that is the average (chunk 2). In the next section we describe how the microplanner manages this organization of the information.

3 The SentenceDesigner microplanner

In a previous version of our system we assumed that the semantic network encoded a single chunk of meaning expressing the semantics of the event only in terms of predicate-arguments. The working hypothesis was to assume a one-to-one sentence alignment between source and target sentences. This simplification assumption allowed for a trivial generation architecture that did not have a microplanning phase at all, and just delegated a simple form of lexicalization to the realizer. However, newer version of the semantic interpreter produced more complex semantic networks. Therefore, in our project we remove the previous assumption and in this Section we describe *SentenceDesigner*, a rule-based microplanner. SentenceDesigner basically performs the following three-steps algorithm:

```
1. Segmentation
   a. Split the semantic network
      into atomic messages
2. Lexicalization
   For each message:
   a. Introduce prelexical nodes
   b. Introduce syntactic relations
      between prelexical nodes
3. Simplification
```

```
For each message:
a. Extend syntactic relations
   among messages
b. Remove non-necessary
   prelexical nodes
c. Remove repetitions
   among messages
d. Remove semantic relations
   and reorder messages
```

In the first step SentenceDesigner split the semantic networks into a number of subgraphs: the idea is to recognize which parts of the network contain an atomic message, i.e. a complete information chunk, that can potentially be generated as a singular sentence. SentenceDesigner uses a very simple heuristic for this step: a message is a subtree of the network, i.e. a root-node together with all of its descendants in the network. We call root-node a node that does not have any parent: in Figure 2 the nodes COMPARISON-RELATION, APPLIED-FUNCTION are root-nodes. Note that some nodes belong to several distinct messages: for example the MATH-VALUE belongs to the messages rooted by COMPARISON-RELATION and APPLIED-FUNCTION respectively.

```
(defrule rule-COMPARISON-RELATION ()
 (semantic-state    (name ££COMPARISON-RELATION) (arg-1 ?X1))
 (semantic-relation (name &COMPAR-ARG1) (arg-1 ?X1) (arg-2 ?X2))
 (semantic-relation (name &COMPAR-ARG2) (arg-1 ?X1) (arg-2 ?X3))
 (semantic-relation (name &COMPAR-OP)    (arg-1 ?X1) (arg-2 ?X4))
   =>
 (assert (syntactic-relation (name SYN-SUBJ) (arg-1 ?X4) (arg-2 ?X2)))
 (assert (syntactic-relation (name SYN-OBJ)   (arg-1 ?X4) (arg-2 ?X3))))

(defrule rule-APPLIED-FUNCTION ()
 (semantic-state    (name ££APPLIED-FUNCTION) (arg-1 ?X1))
 (semantic-relation (name &FUNCTOR) (arg-1 ?X1) (arg-2 ?X2))
 (semantic-relation (name &APPLIED-FUNCTION-VALUE) (arg-1 ?X1)
                                                    (arg-2 ?X3))
   =>
 (assert (syntactic-relation (name SYN-RMOD)  (arg-1 ?X3) (arg-2 ?X2))))
```

Figure 3: Two rules of the knowledge-base used by the expert system for lexicalization.

In the second step, that corresponds to "lexicalization" (Reiter and Dale, 2000), SentenceDesigner performs two distinct procedures for each message. The procedure 2-a. introduces new prelexical nodes in the message that will be treated as lexical items in the realization phase. Also in this case we have a very simple heuristic that associates one-to-one prelexical nodes to concepts and instances. The prelexical nodes are organized into a lexical ontology that is shared with the realizer: in this way the microplanner informs the realizer of the selectional restrictions that the semantics imposes on the syntactic behaviour of lexical nodes (e.g. collocations). For example, the prelexical node value belonging to the class `evaluable-entity` is introduced in place to the concept MATH-VALUE. Note that currently we are not yet able to deal with referring expressions generation for instances, i.e. we uniformly treat concepts and instances: in future we plan to integrate into the system a specific module for this task. The procedure 2-b. concerns the introduction of syntactic relations between prelexical nodes. This is a very complex and critical task: on the one hand we need to encode the linguistic knowledge produced by the corpus analysis (see below) and by many discussions with linguists; on the other hand we need to account for the behaviour of these relations in the CCG used by the realizer. In order to manage this complexity we decided to use an expert system (Stefik et al., 1982).[2] Indeed, expert systems allow for a sharp modularization of the knowledge and allow for a clear resolution of conflicts: we needed several revisions of our formalization and expert systems speed-up this process. In Figure 3 we show two rules that are "fired" by SentenceDesigner during the microplanning of the semantic network in Figure 2: the first rule encodes the *comparison* semantic relation into one *subject* (SYN-SUBJ) and one *object* (SYN-OBJ) syntactic relations; the second rule encodes the semantic relation concerning a mathematical value as a *modifier* (SYN-RMOD) relation. The actual implementation of the system consists of about 50 rules and very complex rules are necessary for particular syntactic constructions as coordination or subordinate clauses, i.e. to manage aggregation.

The third step of the algorithm concerns the simplification of the messages built in the previous step. In 3-a. we "propagate" the syntactic relations among the various messages: if a prelexical node belongs to various messages, then all the syntactic relations starting from that node will be replicated in all the messages. For example, the prelexical node average is replicated in the message rooted by the node COMPARISON-RELATION, since value is connected to the prelexical node average by the syntactic re-

[2] In particular, since SentenceDesigner is written in lisp, we used the LISA expert system. This is an implementation of the RETE algorithm compliant with Common lisp Specifications (Young, 2007).

lation modifier in the message rooted by the node APPLIED-FUNCTION. In 3-b., we remove non necessary prelexical nodes: corpus analysis showed that LIS often is "lexically simpler" with respect to the corresponding Italian sentence, and in order to produce fluent LIS sentences we need to remove some prelexical nodes. For example, the Italian phrase "valori di temperatura" (*values of temperature*) is translated by omitting the sign for "valore". In 3-c., we remove messages that are properly included in other messages: this can happen as a consequence of the procedure 3-a. For example, at this stage the syntactic information of the message rooted by the node APPLIED-FUNCTION is properly contained in the message rooted by the node COMPARISON-RELATION. In 3-d., we remove the semantic relations and reorder the remaining messages on the basis of a simple heuristics: for example, temporal information will be passed first to the realizer. The final

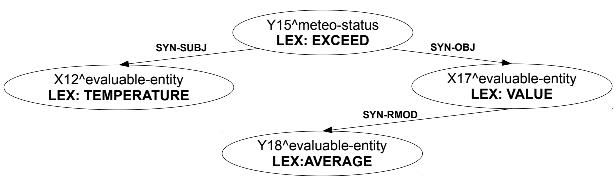

Figure 4: A fragment of the output of SentenceDesigner on the by the semantic network of Figure 2.

result of SentenceDesigner consists of a number of syntactic messages, i.e. a number of abstract syntax trees: each tree will be realized as single sentence (Reiter and Dale, 2000). In Figure 4 there are the abstract syntax tree produced by SentenceDesigner on the input given by the semantic network of Figure 2.

4 A CCG for LIS

In our architecture we use the OpenCCG realizer (White, 2006), an open source tool that is based on categorial grammars (CCG) (Steedman, 2000). Some previous works on translation to SL accounted for typical syntactic phenomena by using lexicalized grammars and feature unification too (Veale and Conway, 1994; Zhao et al., 2000; Huenerfauth, 2006). However we use the OpenCCG since it allows us to encode the LIS inflectional system by using features in the syntactic categories. The

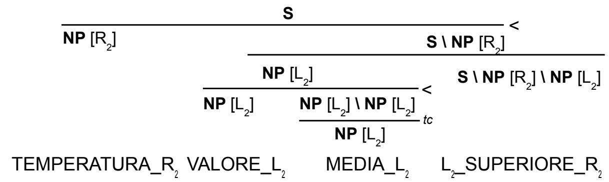

Figure 5: The realization of the LIS sentence "TEMPERATURA_R_2 VALORE_L_2 MEDIA_L_2 L_2_SUPERIORE_R_2".

integration in one single elementary structure of morphology-syntax-semantics is appealing for SLs, where the absence of function words increases the importance of morpho-syntactic features to express the correct meaning of the sentence.

A challenging requirement of our project is that the SLs do not have a *natural* written form. As a consequence we developed an *artificial* written form for LIS. Our electronic lexicon is stored into a database, such that an entry consists of a unique alphanumeric ID. However, for the sake of clarity here we write a LIS sentence just as a sequence of *glosses*. We use names (in uppercase) for the glosses that are related to their rough translation into Italian. The only feature that we explicitly represent in glosses is the *spatial position* of the sign (cf. (Zhao et al., 2000)). We assume a discrete horizontal dimension consisting of seven positions L_1 (the leftmost position), L_2, L_3, N (the neutral position), R_3, R_2, R_1 (the rightmost position).

Similarly to American SL, in LIS we can tell a number of verb classes on the basis of spatial accord (Volterra, 2004; Wright, 2008; Brentani, 2010). For instance the verb L_i_SUPERIORE_R_j (*exceed*) belongs to the class II-A, i.e. it is a transitive verb such that the starting position of the sign (L_i) coincides with the position of the subject, as well as the ending position of the sign (R_j) coincides with the position of the object (Volterra, 2004). Similarly to (Wright, 2008), we model LIS linguistic phenomenon in CCG by using a morphological feature. This feature encodes the position of the noun in the atomic category NP, as well as the starting and ending position of a verb in the complex category $S\backslash NP\backslash NP$ (in accord with (Geraci, 2004) and in contrast to (Volterra, 2004) we assume that LIS respects the SOV order). In Fig. 5 we show the realization of the LIS sentence "TEMPERATURA_R_2 VALORE_L_2 MEDIA_L_2 L_2_SUPERIORE_R_2" by using the abstract syntactic tree in Figure 4. The feature unification mechanism constraints the NP arguments to agree with the starting and ending position of the verb: the subject TEMPERATURA is signed in the position R_2, i.e. the starting position of the verb SUPERIORE, while the object MEDIA is signed in the position L_2, i.e. the ending position of the verb. More details about our formalization of verb-arguments and NP-coordination in LIS can be found in (Mazzei, 2011).

5 Conclusions

In this paper we have presented a generator for LIS adopted into a symbolic translation architecture. The generator is composed by a expert-system based microplanner and a CCG based realizer. The expert-system allows us to manage and update the knowledge provided by linguists and derived from corpus analysis. CCG allowed for a clear formalization of LIS syntax.

While the design of a quantitative evaluation of the system is still in progress, a preliminary qualitative evaluation provided us some information. In particular, two native LIS signers give a positive evaluation about the space allocation of the signs but give a negative feedback on modifiers word order.

Acknowledgments

This work has been partially supported by the ATLAS project, that is co-funded by Regione Piemonte within the "Converging Technologies - CIPE 2007" framework (Research Sector: Cognitive Science and ICT).

References

Cristina Bosco and Vincenzo Lombardo. 2004. Dependency and relational structure in treebank annotation. In *Proc. of the COLING'04 workshop on Recent Advances in Dependency Grammar*, Geneve, Switzerland.

Dana Brentani, editor. 2010. *Sign Languages*. Cambridge University Press.

Carlo Geraci. 2004. L'ordine delle parole nella LIS (lingua dei segni italiana). In *Convegno nazionale della Società di Linguistica Italiana*.

Matt Huenerfauth. 2006. *Generating American Sign Language classifier predicates for english-to-asl machine translation*. Ph.D. thesis, University of Pennsylvania.

Leonardo Lesmo, Alessandro Mazzei, and Daniele P. Radicioni. 2011. An ontology based architecture for translation. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, The University of Oxford.

Leonardo Lesmo. 2007. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47, June.

Alessandro Mazzei. 2011. Building a generator for italian sign language. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 170–175, Nancy, France, September. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge University Press, New York, NY, USA.

Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Mark Stefik, Jan Aikins, Robert Balzer, John Benoit, Lawrence Birnbaum, Frederick Hayes-Roth, and Earl D. Sacerdoti. 1982. The organization of expert systems, a tutorial. *Artif. Intell.*, 18(2):135–173.

Tony Veale and Alan Conway. 1994. Cross modal comprehension in zardoz an english to sign-language translation system. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, INLG '94, pages 249–252, Stroudsburg, PA, USA. Association for Computational Linguistics.

Virginia Volterra, editor. 2004. *La lingua dei segni italiana*. Il Mulino.

Michael White. 2006. Efficient realization of coordinate structures in combinatory categorial grammar. *Research on Language and Computation*, 2006(4(1)):39—75.

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.

David E. Young. 2007. The Lisa Project. http://lisa.sourceforge.net/.

Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman I. Badler, and Martha Palmer. 2000. A machine translation system from english to american sign language. In *Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, AMTA '00, pages 54–67, London, UK, UK. Springer-Verlag.