Lexical Entry Templates for Robust Deep Parsing

Montserrat Marimon* **and Núria Bel**†

*Grup d'Investigació en Lingüística Computacional Universitat de Barcelona
Adolf Florensa, s/n. 08028 Barcelona
montse@gilc.ub.es

†Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra
La Rambla, 30-32. 08002 Barcelona
nuria.bel@upf.edu

Abstract

We report on the development and employment of lexical entry templates in a large–coverage unification–based grammar of Spanish. The aim of the work reported in this paper is to provide robust deep linguistic processing in order to make the grammar more adequate for industrial NLP applications.

1. Introduction

Deep Natural Language Processing (NLP) systems show two main problems: inefficiency processing and lack of robustness for dealing with unknown words and linguistic structures that fall beyond the coverage of the grammatical resources. So, even though they produce a fine–grained analysis of the processed input sentences, as required in those applications where accurate interpretation is needed, such systems are inadequate for real–world applications. In addition, these system often lack methods to select the correct parses when overgeneration is produced (cf. (Grover and Lascarides, 2001; Oepen et al., 2002)).

This paper reports research on the development of lexical entry templates in a large–coverage unification–based grammar of Spanish to address the problem of lack of lexical robustness. We present a hybrid NLP method which interfaces the information delivered by shallow processing components with a set of lexical entry templates in order to provide robust deep processing, while keeping overgeneration up to a reasonable level.

The methodology we propose extends and improves previous proposals within the ALEP framework to obtain more robust (and efficient) deep processing (Bredenkamp et al., 1996; Declerck and Maas, 1997), as well as further related techniques proposed in the literature (Horiguchi et al., 1995; Mitsuishi et al., 1998; Grover and Lascarides, 2001; Crysmann et al., 2002).

The paper is organized as follows. We will start with a brief description of the unification–based grammar that served as the basis of our research work. Then, in section 3, we will present the lexical entry templates we have implemented, and we will show how we avoid overgeneration without losing coverage. Section 4 reports on an experiment which measures the performance of the lexical entry templates in the ALEP system. Finally, section 5 presents the general conclusions.

2. Grammar Overview

Grammar development was done in the framework of the Advanced Language Engineering Platform (ALEP) (Simpkins et al., 1993) during the European projects LS–GRAM (LRE–61029) (Schmidt et al., 1996), MELISSA (ESPRIT–22252) (Bredenkamp et al., 1998) and, more recently, IMAGINE (IST–2000–29490) (Arana et al., 2003). Both in MELISSA and in IMAGINE the grammar was used in an industrial context.

It is indeed a large–scale grammar whose coverage has been defined on the basis of corpus investigations, and it copes with input ranging from short instructive statements or queries —including non–sentential input strings— that appeared in the corpus of the MELISSA and the IMAGINE applications[1] to complex sentential structures as are found e.g. in newspaper articles.

The adopted approach in the grammar basically follows HPSG proposals (Pollard and Sag, 1994). It is a highly lexicalized grammar where the lexical component plays a crucial role in the grammatical description needed for linguistic processing, and where grammar rules are reduced to a small set of binary–branching context–free phrase structure rules which implement the universal linguistic principles governing HPSG ID schemata. Both lexical entries and grammar rules are based on a type system that constitutes a monotonic simple type hierarchy with appropriateness conditions.

The grammar produces a complete syntactic and semantic analysis of the sentences it processes, however, it fails in producing a result when the words —and the linguistic structures being processed— fall beyond the coverage of the grammar.

[1] Both MELISSA and IMAGINE aimed at developing technology for being used in natural language interface applications. In MELISSA, the deployment scenario was ICAD, an administrative purchase and acquirement handling system, used at the Organización Nacional de Ciegos de España (ONCE). In IMAGINE, the deployment scenario was the Viapolis wap application, an online network delivering local entertainment, commerce, news and community resources.

3. Lexical Entry Templates

To provide robust deep linguistic processing we have implemented default lexical entries, i.e. lexical entry templates that are activated when the system cannot find a particular lexical entry to apply.

Basically, there are two ways to define default lexical entries. One is to define underspecified lexical entry templates assigned to content words (nouns, verbs, adjectives, adverbs) in such a way that, while parsing, the system fills in the missing information in the lexical entry template of each unknown word by the application of phrase structure rules (or rule schemata and principles, in HPSG–based grammars) (Horiguchi et al., 1995; Mitsuishi et al., 1998; Grover and Lascarides, 2001; Crysmann et al., 2002). In the other approach, very detailed default lexical entries for each content word class are defined.

The strategy we have followed falls under a middle type, and we have implemented default lexical entries covering the most frequent subcategorization frames of each major word class, i.e. on the basis of both the category and the number of subcategorized for elements (subjects and complements). These default entries, however, are underspecified with respect to the semantic features encoding selectional restrictions imposed on subcategorized for elements, which are specified by the application of grammar rules.

The use of lexical entry templates in a highly lexicalized grammar, however, may increase ambiguity, and, thus, overgeneration (and processing times). In order to reduce ambiguity, (Mitsuishi et al., 1998) added additional non–linguistic constraints to the original grammar components (ID schemata, lexical entries and lexical entry templates). Such constraints disallow the treatment of rare linguistic phenomena, but they are too strong, and, therefore, cause some coverage loss.[2]

To restrict as much as possible the templates that are activated and to keep overgeneration up to reasonable level, while maintaining the coverage of the grammar, we propose a hybrid architecture (Figure 1).

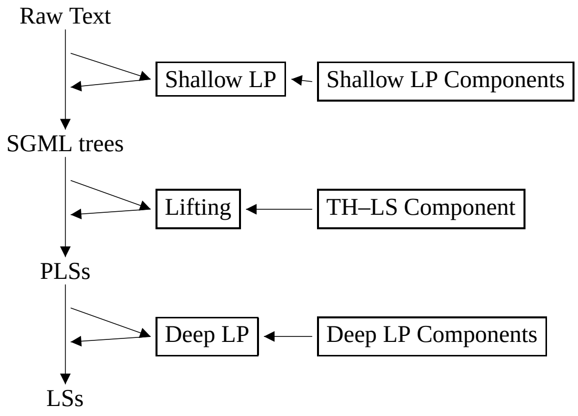

Figure 1: The hybrid architecture.

We have integrated a cascade of shallow linguistic processing components performing mark–up of special constructions (or named entity recognition) e.g. dates, numbers, proper names, etc., morphological analysis, PoS disambiguation and shallow parsing (chunking), in the ALEP environment. Note that the success of a hybrid system is significantly dependent on the performance of the shallow processing components. An analysis of the failures of the system described by (Grover and Lascarides, 2001) revealed that a bit more of the 60% of failures of the methods they propose were due to preprocessing (segmentation and tagging) errors. Therefore, we integrated a linguistic tagger (as opposed to data–driven tagger) that leaves ambiguities to be solved by the following deep linguistic processing components rather than making risky predictions.[3]

As can be observed in Figure 1, the output of the shallow processing components, which consists of a SGML–based marked–up tree, is converted into Linguistic Description (LD) ALEP data types, modeling lexical entries and structure nodes, in a non–immediate dominance relation, expressing the hierarchical relations between the different structural elements. This conversion is performed by a set of so–called Text Structure to Linguistic Structure (TS–LS) rules which allow the flow of information between the attribute–value pairs of the SGML–tags and the attribute–value pairs of target LDs (Figure 2). Finally, deep LP components build up a parsed tree (or a Linguistic Structure (LS) ALEP data type) from which we obtain compositional meaning representations.

ts_ls_rule(

$$\left[\begin{array}{l} \textit{ld} \\ \text{SYNSEM}\,|\,\text{LOC}: \left[\begin{array}{l} \textit{t\_local} \\ \text{MORPH}: \left[\begin{array}{l} \textit{t\_morph} \\ \text{LEMMA}: \boxed{2} \\ \text{MORPHEME}: \boxed{1} \\ \text{AGR}: \textit{(fem\&sing)} \end{array}\right] \\ \text{CAT}: \left[\begin{array}{l} \textit{t\_subst} \\ \text{HEAD}: \left[\begin{array}{l} \textit{t\_noun} \\ \text{NCLASS}: \textit{common} \\ \text{FRAME}: \left[\begin{array}{l} \textit{t\_frame} \\ \text{FTYPE}: \boxed{3} \\ \text{MK\_PREP}: \boxed{4} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right],$$

$[\ldots, \texttt{POS = 'Ncfs-'}, \texttt{LM =} \boxed{2}, \texttt{FT =} \boxed{3}, \texttt{MP =} \boxed{4}], \boxed{1}).$

Figure 2: TS-LS rule lifting nominal tag

Lexical entry templates are activated on the basis of the information encoded in the lexical tags delivered by the shallow processing tool which, as can be observed in figure 2, include not only morpho–syntactic information, but also syntactic information about the subcategorized for elements, including: the category and number of complements, marking prepositions, mood and form of finite verbal complements, and information about valence changing

[2] (Marimon and Bel, 2003) show a method to integrate shallow parsing which provides deep NLP with larger–coverage for rare (and, even, ungrammatical) syntactic structures while avoiding ambiguity problems when parsing well–formed sentences.

[3] Some statistics on the results on the performance of the tagger we used may be found in (Marimon, 2003).

operations (movement and/or removal of complements).[4]

Our strategy requires some minor changes in the ALEP lexicon: the introduction of new features encoding the framing information (encoded in the feature FRAME of figure 2) that is delivered by the shallow processing components. This redundant information, which is eliminated by the application of phrase structure rules (i.e. the feature FRAME is not percolated to structure nodes), does not only ensure that a unique lexical entry template is activated for a given unknown word, and, therefore, improves the accuracy of the grammar performance, but it also avoids TS–LS rule diversification according to the different verbal, nominal and adjectival frames. Note that TS–LS rules, however, are diversified on the basis the POS tag feature —encoding the morphosyntactic information— takes, since in the ALEP system value–sharing between SGML–tag and LD levels of representations is only allowed for atom–valued features of LDs.

Figure 3 partially shows the lexical entry template we defined for non–modifying non–predicative common nouns.

Besides providing robustness to the deep processing components, and similarly to (Grover and Lascarides, 2001; Crysmann et al., 2002), lexical tags are also used to select the correct reading of lexical entries listed in the ALEP lexicon to reduce the ambiguity of the linguistic expression to be analyzed, making the system perform significantly better.

Our proposal extends previous hybrid proposals within the ALEP framework to obtain more robust (and efficient) processing. (Bredenkamp et al., 1996) describe how, in the context of the LS–GRAM project, language–specific taggers were developed and integrated into the text handing ALEP component for the recognition and mark–up of special constructions. Also in the context of the LS–GRAM project, (Declerck and Maas, 1997) extend the functionality of this external add–on module and use it to integrate into the ALEP processing components part–of–speech information —the category— delivered by a PoS tagger (the `Mpro` tool (Maas, 1996)) which may be use to reduce lexical ambiguity.

$$\begin{bmatrix} \textit{ld} \\ \text{SYNSEM}\,|\,\text{LOC}: \begin{bmatrix} \textit{t\_local} \\ \text{MORPH}: \begin{bmatrix} \textit{t\_morph} \\ \text{LEMMA}: \boxed{1} \\ \text{AGR}: \boxed{2} \end{bmatrix} \\ \text{CAT}: \begin{bmatrix} \textit{t\_subst} \\ \text{HEAD}: \begin{bmatrix} \textit{t\_noun} \\ \text{AGR}: \boxed{2} \\ \text{MOD}: \langle\rangle \\ \text{FRAME}: \begin{bmatrix} \textit{t\_frame} \\ \text{FTYPE}: a \\ \text{MK\_PREP}: x \end{bmatrix} \end{bmatrix} \\ \text{SUBJ}: \langle\rangle \\ \text{COMPL}: \langle\rangle \end{bmatrix} \\ \text{SEM}: \begin{bmatrix} \textit{t\_sem} \\ \text{INDEX}: \textit{sf(index(nevent,}\boxed{1}\textit{))} \\ \text{PREDARG}: \begin{bmatrix} \textit{t\_predarg} \\ \text{PRED}: \textit{sf(pred(}\boxed{1}\textit{))} \\ \text{ARGS}: \langle\rangle \end{bmatrix} \\ \text{MOD}: \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 3: Lexical entry template

4. Evaluation

The evaluation of the performance of the lexical entry templates in the ALEP system was done with free input text. We used a newspaper article of 268 words.

Even though about 68% of content words in the article were unknown to the system (i.e. were not encoded in the lexicon) —46% of the verbs, 78% of the nouns, 50% of the adjectives, 50% of the adverbs—, the system did not fail in producing a result because of lack of lexical coverage.

Some statistics on results are given in table 1. The second column shows the average of analysis we get by activating lexical entry templates only on the basis of the morphosyntactic information. The third column shows how overgeneration is reduced by also using the framing information delivered by the shallow processing components.

unknowns words	Version 1	Version 2
67.7%	8.2	2.2

Table 1: Results

5. Conclusions

We have described the development and employment of lexical entry templates in a large–coverage unification–based grammar of Spanish to provide robust deep linguistic processing.

Even though the strategy we have outlined in the previous sections is largely dependent on the grammar development environment we used, we expect it to be applicable within other deep NLP systems.

[4] The notation format adopted in the lexical tags consists of a numbered string of characters —attributes are marked by positions— where: (i) the first character encodes part–of–speech, and (ii) the following characters encode the value of one attribute relevant for each category. To encode the different frames the strategy we followed is inspired by EUROTRA (EUROTRA, 1991), where frames are coded by a single letter, e.g. 'a' for verbs taking nominal subjects, 'b' for verbs taking nominal subjects and indirect objects, 'c' for verbs taking nominal subjects and direct objects, etc. and where letter can be combined compositionally to deal with frame alternations. Thus, a verb taking a nominal subject and an optional direct object takes the value 'ac', whereas a verb taking taking a nominal subject, a direct object and an optional indirect object takes the value 'ce'. If an attribute is not relevant for lexical item, the corresponding position is underspecified (i.e. it takes '-' as value). Unspecification of one value (i.e. when all values may be relevant for a given lexical item) is expressed by a dot'.'. Finally, combined letters appear between square brackets, such that they only occupy a position in the tag.

6. References

Arana, C., I. Recio, M.J. Carrin, J. de Frutos, I. Dattani, F. Choi, P. Wilken, S. Hefeez, M. Cecil-Wright, P. Schmidt, M. Marimon, C. Pease, J. Hinz, V. Caminero, D. Castell, L. Hernandez, J. Relao, K. Gladstone, R. Pick, E. Ramos, J. Alicarte, and A. Pons, 2003. IMAGINE: Interfacing Mobile Application with Voice Natural Language Interactivity. In *Procesamiento del Lenguaje Natural*, number 31.

Bredenkamp, A., T. Declerck, P. Groenendijk, M. Phelan, S. Rieder, P. Schmidt, H. Schulz, and A. Theofilidis, 1998. Natural language access to software applications. In *Proceedings of the Joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING–ACL'98)*. Montreal, Canada.

Bredenkamp, A., F. Fouvry, T. Declerck, and B. Music, 1996. Efficient integrated tagging of word constructs. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING–96)*. Copenhagen, Denmark.

Crysmann, B., A. Frank, B. Kiefer, H.-U. Krieger, S. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, and F. Xu, 2002. An integrated architecture for shallow deep processing. In *Proceedings of Association for Computational Linguistics 40th Anniversary Meeting (ACL–2002)*. University of Pennsylvania, Philadelphia, PA.

Declerck, T. and H.D. Maas, 1997. The integration of a part–of–speech tagger into the ALEP platform. In *Proceedings of the 3rd ALEP User Group Workshop*. Saarbrücken, Germany.

EUROTRA, 1991. *Spanish Final Implementation Report*. Commission of the European Communities.

Grover, C. and A. Lascarides, 2001. XML–based data preparation for robust deep parsing. In *Proceedings of the Joint EACL–ACL Meeting (ACL–EACL 2001)*. Toulouse, France.

Horiguchi, K., K. Torisawa, and J. Tsujii, 1995. Automatic acquisition of content words using an HPSG–based parser. In *Proceedings of the Natural Language Processing Pacific Rim Symposium*. Seoul, Korea.

Maas, H.D., 1996. MPRO - ein System zur Analyse und Synthese deutscher Wörter. In R. Hausser (ed.), *Linguistische Verifikation, Sprache und Information, Nr. 34*. Tübingen, Germany: Max Niemeyer Verlag.

Marimon, M., 2003. *On Distributing the Analysis Process of a Broad–Coverage Unification–based Grammar*. Barcelona: PhD thesis, Departament Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Marimon, M. and N. Bel, 2003. A Hybrid NLP for NLIs. In *Proceedings of International Conference Recent Advances in Natural Language Proceecessing (RANLP-2003)*. Borovets, Bulgaria.

Mitsuishi, Y., K. Torisawa, and J. Tsujii, 1998. HPSG–style underspecified Japanese grammar with wide coverage. In *Proceedings of the Joint 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING–ACL'98)*. Montreal, Canada.

Oepen, S., E. Callhan, D. Flickinger, C.D. Manning, and K. Tautanova, 2002. LinGO Redwoods. A Rich and Dynamic Treebank for HPSG. In *Proceedings of LREC Workshop: Beyond PARSEVAL: Towards Improved Evaluation Measures for Parsing Systems. Third International Conference on Language Resources and Evaluation (LREC–2002)*. Las Palmas de Gran Canaria, Spain.

Pollard, C. and I.A. Sag, 1994. *Head–Driven Phrase Structure Grammar*. Chicago: University of Chicago Press and CSLI Publications.

Schmidt, P., S. Rieder, A. Theofilidis, and T. Declerck, 1996. Lean formalism, linguistic theory, and applications: grammar development in ALEP. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING–96)*. Copenhagen, Denmark.

Simpkins, N. K., M. Groenendijk, and G. Cruickshank, 1993. *ALEP User Guide*. Luxembourg: Commission of the European Communities.