A modular architecture for the processing of free text

Toni Badia, Gemma Boleda, Martí Quixal, Eva Bofias
Institut Universitari de Lingüística Aplicada
Universitat Pompeu Fabra
Rambla, 30-32 Barcelona E-08002
toni.badia@trad.upf.es,{gemma.boleda,marti.quixal}@iula.upf.es,
eva@RhetoricalSystems.com

To appear in the Proceedings of the Workshop on
'Modular Programming applied to Natural Language Processing' at EUROLAN 2001

Abstract

This paper describes the free text processing strategy that is being set up in our institute. The system is designed to deal with general, written Catalan texts, as they appear in, say, daily newspapers. Our strategy has been to divide the whole processing into specific subtasks, applying to each of them the best strategy available. The main advantages of the architecture we put forth are that it is highly modular and reusable, and that it permits a fully automatic processing of unrestricted text.

1 Introduction

The processing streamline that we envisage is intended to carry out the automatic analysis of real Catalan texts. From the start it has been designed in a modular way, so that the best strategy for each specific task can be chosen, and a progressive improvement of the whole processing can be obtained as new modules are available.

We are interested in the tagging of texts with linguistic information, so that the operations that are performed on them can be based not only on their surface form but also on their linguistic structure. Our aim is to achieve a linguistic tagging of running text as precise and detailed as possible, bearing in mind a wide range of possible further applications (from grammar checking to information extraction). This tagging involved initially only part-of-speech, but is being extended to morphosyntactic and strictly syntactic information. We also plan to include semantic and pragmatic information in the future.

It is however impossible to achieve this complex task in one shot, since neither the resources nor the techniques are fully available at one given moment in time. We therefore developed a processing setting in which we could (1) start processing and extracting information from texts from the very beginning of the project; and (2) add new modules if and when they were available.

The paper is organised as follows: section 2 describes the basic architecture of the system: the text handler and the morphological and syntactic analysis modules. Section 3 describes how we took advantage of previous existing tools (created at our institute or not). Section 4 presents several modules that we plan to add to our parsing architecture to achieve a deeper analysis. Section 5 details the current state of the project. Section 6 is a comparison with other approaches. The paper ends with some conclusions.

2 Basic architecture

At the beginning of the process (see Figure 1) we have a small text handling module, that prepares the text to be tagged with linguistic information. The kernel of the processing schema is the set of the modules covering the morphological and shallow syntactic analysis. In each case there is a distinction between the initial assignment of tags and the subsequent disambiguation.

In the following subsections we describe each of the modules separately.

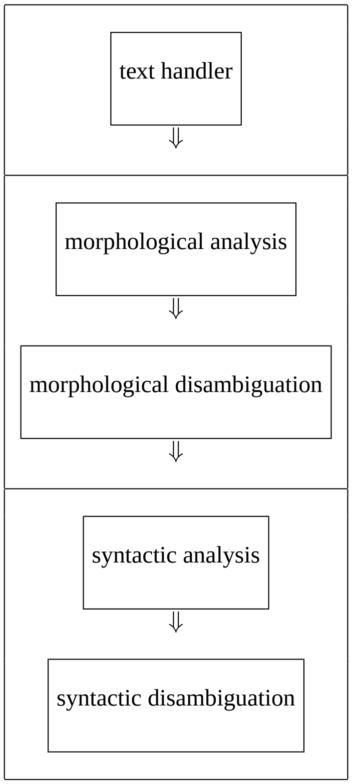

Figure 1: Kernel of the processing schema

2.1 Text handler

As shown in Figure 1, the first stage of our process is performed by a text handler that structures raw text by means of SGML tags and performs the following tasks:

- Sentence delimiting: identification of paragraphs and sentences, handling abbreviation and punctuation and distinguishing headlines from ’normal’ paragraphs.

- Word separation and verticalisation. This involves separating elements contained in contracted, apostrophed and hyphenised constructions, frequent in Catalan[1].

- Figure and date identification.

In Figure 2 we present an example sentence that we will use throughout the article (for space purposes, we display this sentence ’horizontalised’, not verticalised as it should look after this first substage).

2.2 Morphological analysis[2]

Before we start detailing our morphological module, there are some theoretical and practical aspects that characterise the whole morphological process we would like to describe.

The first one is the tag set, created in our institute in 1997, see (Morel *et al.* 1997) for details, and elaborated following some of the standards proposed by the EAGLES project. The criteria used were both functional and linguistic, for we wanted as descriptive and theoretically sound a tag set as possible, but with the minimal complexity, for computational reasons.

The total number of basic morphological categories of the tag set amounts to 25. Each category may have attributes like, for instance, gender and number in nouns. If we count each of the categories with each one of its possible attribute combination, we obtain approximately 350 tags (which express more information than we can extract to date from electronic lexical bases). Actually, only about 200 are being used.

The tags may be visualised in a contracted or expanded manner depending on the module’s requirements (see both appearances in Figure 3). They may consist of up to 7 character positions (each one containing a relevant piece of information if available, and the first one being always the morphological category). Several tags allow underspecification of information whenever ambiguities happen to be systematic. For instance, the tag E (specifier) is used for all those words that behave like a determiner or an adjective and are able to pronominalise (as is the case in Catalan for most quantifier, possessive, cardinal, ordinal and indefinite determiners).

Our module uses three different programming devices (2 coding languages and a grammar-writing framework). On the one hand, we use a Prolog-based word form analyser/generator and several Perl scripts for the tag mapping process. On the other hand, we are developing three different rule-based grammars created within the Constraint Grammar (CG) framework ,see (Karlsson *et al.* 1995) and (Tapanainen, 96) for morphological and syntactic disambiguation, and for syntac-

[1] Units such as adverbial, prepositional and conjunctive locutions are better analysed as morphosyntactically independent words. Its treatment as units only makes sense in later stages.

[2] The reader may have noticed that we use the term morphological analysis in its broad sense. That is, it involves two subtasks: morphological tag mapping and disambiguation.

$< p >< s >$	La	casa	$\acute{e}$s	verda.	$< /s >< /p >$
$< p >< s >$	the-FEM-SG	house-FEM-SG	is	green-FEM-SG.	$< /s >< /p >$

Figure 2: Sentence after the morphological tag mapping

tic mapping.

The CG formalism is an ideal tool to achieve a certain level of analysis previous to a deeper linguistic parsing (see section 4). In (Karlsson *et al.* 1995):1 it is stated that CG is 'a language-independent formalism for surface-oriented, morphology-based parsing of unrestricted text. [...] All relevant structure is assigned via [...] simple mappings from morphology to syntax. The constraints discard as many alternative as possible [...] with the proviso that no genuine ambiguities should be obliterated'.

2.2.1 Morphological tag mapping

This step has recently undergone a major change: instead of being performed by a word form analyser in run-time, as used to be the case, it is now realized by a simple word form dictionary. The dictionary is generated by the old morphological analysis tool, which has been converted into a form generator. Thus we have increased speed and still profit from the advantages of the old module (see 3.2 for details).

As for the process, note that the morphological mapping is not context-sensitive (in contrast to the syntactic mapping; see section 2.3.1), that is, each word form is assigned every possible reading according to the information read in the dictionary.

An important characteristic of our morphological mapping is the fact that it provides partial subcategorisation information for verbs, once their lemmata have been identified. This improves both morphological and syntactic disambiguation (in angle brackets in Figure 3).

The result of this process, following CG terminology, is a text with cohorts (that is, word forms and all their possible readings) as shown in Figure 3. The reader should notice that the two first words are ambiguous: in the next stage we will get rid of the pronoun (Pron) reading for *la* and the verb readings for *casa*.

2.2.2 Morphological disambiguation

As mentioned above, we have developed a CG-based morphological disambiguation engine for Catalan (*Desambiguador Morfològic per al Català*, DeMCat) with over 1000 rules. The basic strategy is to select or remove certain tags according to the constraints imposed by the surrounding context. The appearance of a rule is the following:

OPERATOR (TARGET-TAG) IF (CONTEXT)

The TARGET-TAG is the tag on which the rule is going to operate. The OPERATOR indicates whether the target tag is going to be selected or removed. The CONTEXT specifies the surrounding words / tags needed in order for the rule to apply. Context positions are indicated with positive (right of target) or negative (left of target) integers. The CG formalism provides other devices, like Kleene's star, the possibility to work with relative or absolute positions, or care modes to tune the rule application. It also makes it possible to use heuristic disambiguation by means of weighted rules.

The result of this process will be a (partially) disambiguated text (see Figure 2.2.2). In our example, there are now three less morphological tags: the Pron reading was eliminated from *la* and two Verb tags were eliminated from *casa*. The rule that applied for the elimination of the Verb readings of *casa* was the following:

REMOVE (Verb) IF (0 NOM + FS) (-1 MODI + FS) (*1C VFIN)

This rule states that the reading Verb should be discarded from words that can be nouns (NOM), provided they have a modifier (MODI) agreeing in gender on their left and an unambiguous finite verb (VFIN) anywhere on their right. In our example, the two Verb readings for *casa* have been removed, since *casa* has *la* on its left, which is a feminine singular determinant, and *és* on its right, which is a finite verb form.

```
“< La >”
                “el” Det fem sg
                “lo” Pron person febl acus 3pers fem sg REEC3FS
“< casa >”
                “casa” Nom com fem sg N5-FS
                “casar” < S >< o >< Ps >< NA > Verb MInd Pres 3pers sg VDR3S-
                “casar” < S >< o >< Ps >< NA > Verb MImp Pres 2pers sg VRR2S-
“< és >”
                “ser” < SS >< A > Verb MInd Pres 3pers sg VDR3S-
“< verda >”
                “verd” Adj qual fem sg JQ–FS
```

Figure 3: Sentence after the morphological tag mapping

```
“< La >”
                “el” Det fem sg
“< casa >”
                “casa” Nom com fem sg N5-FS
“< és >”
                “ser” < SS >< A > Verb MInd Pres 3pers sg VDR3S-
“< verda >”
                “verd” Adj qual fem sg JQ–FS
```

Figure 4: Sentence after the morphological disambiguation

2.3 Syntactic analysis[3].

The two most important features of our syntactic analyser are: (1) it is a surface-oriented parser; i.e., it avoids the use of empty categories. And (2) it is a shallow parser; i.e., it does not (completely) deal with constituency. Both are very appropriate characteristics for our step-by-step strategy.

The syntactic analysis provides each word with a tag indicating its syntactic function: it is always a head function (like subject, object or main verb) or a head-dependent function (like noun modifier or determiner, or other *ad hoc* tags such as preposition complement – such tags are used to point to constituency structures. For instance, preposition modifiers would be heads of constituents introduced by a preposition, independent of their morphological tag, and would therefore be assigned the tag @<P. This tag introduces the use of brackets pointing to its phrasal head. This helps further in determining constituency in further steps (see section 4.1).

Practically, the principal function tag is assigned to the main word: for instance, in our example it is *casa* the word that has to be assigned the tag @Subj(ect). On its behalf, *la* will be assigned @DN> (Noun determiner): we can see here that the angle bracket indicates the fact that *la* depends on a subject phrase head.

As for the tag set, it presently amounts to approximately 30 items. It has been created following several traditional grammars, (Fabra 1956) and (Badia i Margarit 1994), the criteria of which have been adapted to make them more functional and theoretically sound.

As for the rules of our syntactic analyser (both the syntactic tag mapping module and the disambiguation one), they look similar to those used for morphological disambiguation (see section 2.2.2). The new thing about them is that the mapping module rules map syntactic tags on morphologically tagged words.

2.3.1 Syntactic tag mapping

The syntactic tag mapping, for which we use a CG-module with around 300 rules, is the first substage of the syntactic analysis. The morphosyntactic tags available allow us to control the pro-

[3] As in the previous section, the term analysis refers here to both syntactic tag mapping and disambiguation

cess, so that some impossible ambiguities are avoided, making the disambiguation task easier.

The following rule illustrates how controlled mapping avoids unnecessary ambiguity in our example:

MAP (@Subj) IF (0 DET) (NOT *1 NOM BARRIER Q_MOT/MGN)

The rule states that determiners (DET) are to be assigned the tag @Subj unless there is a common noun (NOUN) at their right. This is coherent with our linguistic approach, in which determiners are heads of subjects unless they are specifying a noun. As can be seen in Figure 5, it does not apply to our sentence, because *la* (a determiner) had *casa* (a noun) at its right side.

2.3.2 Syntactic disambiguation

Our CG-based syntactic disambiguation module has currently about 1400 rules. The strategy adopted is to remove as many readings as possible, and rely only on tag selection in very specific contexts. For this task, we use the morphological information available from the previous steps, together with the progressively obtained syntactic information.

As one might expect, some ambiguities still remain after the application of this module. Some are due to the lack of subcategorisation information in the lexicon. Some other are due to limitations of the formalism, because its surface-oriented approach does not completely account for constituency. PP-attachment exemplifies both problems (see section 4.2 for strategies to deal with these constructions).

Let us exemplify one of the rules that applied in this module:

SELECT (@Subj) IF (0 NOM) (NOT *1 SUBJ) (NOT *-1 SUBJ)

This rule states that a noun should be selected as subject if it has a @Subj tag, and no other elements of the sentence are candidates for this function. As we can see in 2.3.2, this is the case of *casa* in our example, since neither *la* nor *verda* can play such role.

3 Reuse of previously existing modules

3.1 Dictionary extraction

The large computational lexicon we use for word form generation (see section 2.2.1) was semi-automatically built from (DIEC), a Machine-Readable Dictionary, see (Tuells 1998) for details. This MRD is a recent human-reader-in-mind dictionary for Catalan available in electronic form. The information extracted was each headword, its part of speech, and the inflectional paradigm of nouns, adjectives and verbs.

The inflectional paradigm of the words was formally represented as the lexical rules used in the morphology processor (see section 3.2). As for the irregularities, explicit in (DIEC), they were represented as either the blocking of (some of) these rules or a lexicalized word form (in minor cases). Around 68000 lexical entries were automatically added this way, and only around 2800 (800 nouns, 2000 verbs) were added manually. Many other entries have been added, both from other electronic sources (a descriptive dictionary, (DLC)) and -the least- by hand.

3.2 Morphology processor

Initially, the morphological mapping was carried out by a tool developed at our university : CATMORF, a module written in Prolog which was the first wide-coverage two-level morphological analyser for Catalan, see (Badia *et al.* 1998). This tool models morphotactics in a (DCG-like) unification word grammar, and morphographemics in SEGMORF, an extension of the ALEP (Advanced Language Engineering Platform) morphographemic formalism. SEGMORF seeks to deal with morphological phenomena in a way that allows a well-defined, linguistically motivated interaction between the morphographemic and the morphotactical components of the morphological processor, see (Badia & Tuells, 1997).

CATMORF, thus, has been a central module of the tagging process. Its input was the result of the text handler and its output went to the morphological disambiguation module in run-time. We are in the process of changing the system in order to make it faster, more flexible and more manageable: the idea is to use CATMORF as a word form generator instead of as an analyser. We thus ob-

```
“< La >”
                “el” Det fem sg @DN>
“< casa >”
                “casa” Nom com fem sg N5-FS @Subj @Atr
“< és >”
                “ser” < SS >< A > Verb MInd Pres 3pers sg VDR3S- @VPrin
“< verda >”
                “verd” Adj qual fem sg JQ–FS @Subj @CD @Atr
```

Figure 5: Sentence after syntactic tag mapping

```
“< La >”
                “el” Det fem sg @DN>
“< casa >”
                “casa” Nom com fem sg N5-FS @Subj
“< és >”
                “ser” < SS >< A > Verb MInd Pres 3pers sg VDR3S- @VPrin
“< verda >”
                “verd” Adj qual fem sg JQ–FS @Atr
```

Figure 6: Sentence after syntactic disambiguation

tained an inflected form table that can be easily read by a Perl script that maps morphological tags to each word (see section 2.2.1).

This new organisation will spare processing time, for consulting this table will be much faster than running CATMORF. It will also reduce the number of programming devices on which the system depends. Moreover, it will make the process clearer and easier to manage and modify, while we will still benefit from CATMORF as a powerful tool to build and update an extensive lexical database.

4 Extensions to the basic architecture

4.1 HPSG-style grammars

We are developing a strategy for combining the shallow lexical morphosyntactic tagging explained in section 2 with phrase structure syntactic parsing, which reflects constituency and dependency, see (Badia & Egea 2000). This is part of the overall strategy we follow, which consists in splitting the syntactic analysis into different levels (processing steps) in order to improve efficiency without loss of analytic power. The next step would then be to expand the parsing with semantic and pragmatic information.

We are implementing a unification-based grammar in ALEP (Simpkins 1995), which can parse unrestricted text using a very reduced lexicon (about 100 entries). This is possible because the lexicon entries are actually morphosyntactic feature structures (FS), instead of word tokens or word types. Each word is assigned the FS (entry) that matches the information provided by the previous analysis. This procedure makes grammar writing much easier and controllable.

4.2 PP-attachment disambiguation module

We are currently carrying out research in order to take advantage of feedback techniques combining electronic resources (dictionaries, lexical databases, ontologies) and morphosyntactically tagged corpora.

The first step in this direction is the development of a specific PP-attachment disambiguation module that can tackle the difficulties mentioned in section 2.3.2. This module, following the spirit of (Rigau 1998), is being designed so as to encode and exploit various linguistic data. Some of these data, such as information on the derivation process of the word, subcategorisation frames, semantic categorisation, or its semantic relation to other words, is being automatically extracted from an electronic dictionary (DIEC). Aftewards a value called Semantic Relation (SR) will be calculated in order to quantify the previously extracted relations. The final PP-attachment disambiguation tool will use all those pieces of information to actually perform the disambiguation.

5 Current state of the project and results

Our tagging process has been developed during the last six years to tag large corpora in Catalan. Until now, we have used CATMORF as a text analyser (see section 3.2) and a stochastic morphological disambiguator. Therefore, the documents in our corpus have gone through a three-stage architecture: the text handler, the (old) morphological tag mapper and a stochastic morphological tagger.

The architecture we envisage now faces two major improvements: on the one hand, our tagging process is going to be faster (using the word form tables instead of CATMORF) and we are going to use linguistic-based rules for disambiguation (without dismissing the possibility to implement a stochastic disambiguating module afterwards). Furthermore, we want to introduce syntactic surface-oriented parsing in the actual tagging process as a previous stage to a deeper analysis (see section 2.3). We now proceed to a more detailed description of the current state of each module in our new architecture:

- Text handler: completed (except for entity detector).
- Morphological tag mapper (as described in 2.2.1): completed.
- Morphological disambiguator: completed, in evaluation.
- Syntactic tag mapper: completed.
- Syntactic disambiguator: under development (to be finished by the end of the year).

6 Comparison with other approaches

From a general point of view, our processing architecture fits in the modular, as opposed to non-modular, environments. We thus avoid typical problems for the latter kinds of systems. Their major problem is, as widely recognized, its lack of adaptability: that is, a change in a step of the processing task implies reorganizing the whole system, with the cost and the time that this involves. As we have seen throughout the paper, we can (and have done it) substitute or redesign any module without affecting the other ones.

Another important problem in non-modular environments is the detection of error sources, which is much easier in our system, as far as the result of every task can be treated separately. Not to mention the special error detecting mechanisms available through the coding devices we use (trace and debugging mainly).

We now turn to a comparison with another processing architecture (to our knowledge, the only application for Catalan apart from ours), which was developed at UPC (Universitat Politècnica de Catalunya, see (Carmona *et al.* 1998)). In fact, this architecture was developed entirely for Spanish, and afterwards partially adapted to Catalan (for which neither accuracy nor speed data are available). It includes a morphological analyser (*maco+*)[4], two morphological disambiguators and a shallow syntactic parser.

The morphological disambiguaton in the UPC system is performed by either a statistical decision-tree based tagger (*TreeTagger*) or a relaxation labelling based tagger (*relax*), although it is foreseen to combine their results in order to improve accuracy, see (Carmona *et al.* 1998). As for the syntactic parser, it uses a bottom-up chart parser and provides either a chunking of the text or a full parsing. Nevertheless, it must be noted that the information provided by the UPC parser concerns only constituency and phrase type, instead of grammatical function as in our parser (note that both systems are quite complementary in this respect): in our example sentence, which we have tested against the demo available on the UPC web, the system tells us that *la casa* is a NP, but it doesn't say anything about it being a subject. This information cannot be inferred from the tree, since post-verbal NPs are analysed in the same fashion (as NPs depending on the VP-node) even if they are subjects that have been focused (which a very common practice in Catalan).

To sum up, the major difference between the UPC system and ours is the approach cho-

[4] This module is practically equivalent to ours, so we won't comment on it.

sen. Both morphological disambiguators and the parser developed at the UPC are stochastic, while our system is based on linguistic information (although we plan to combine it with heuristics at a later stage, see previous section)[5].

7 Conclusions

We have presented a highly modular parsing architecture, that by the end of the year will be automatically tagging large corpora with morphosyntactic information. The architecture we propose shows the advantages of modularity and reusability.

Modularity is proven by the fact that we can replace our old morphological stochastic tagger with a new linguistic-based one without having to modify the rest of the process. And we still have the possibility to use a new stochastic tagger after the new results.

As for reusability, we have described two procedures in which we took advantage of existing tools: (1) the creation of a morphological analyser and generator and (2) the creation of a word form dictionary out of this last tool.

Finally, we expect to be able to apply feedback techniques (current research), in particular, for the semantic analysis by means of syntactic pattern detection combined with available electronic resources.

References

Badia i Margarit, A.M. (1994) *Gramàtica de la llengua catalana : descriptiva, normativa, diatòpica, diastràtica.* Barcelona: Enciclopdia Catalana.

Badia, T., À. Egea & T. Tuells (1997) CATMORF: Multi-two level steps for Catalan morphology. In *Demo Proceedings of the Conference on Applied Natural Language Processing*. Washington, 1997.

Badia, T. & T. Tuells (1997) On dealing with morphographemic and morphotactical interaction phenomena in SEGMORF. In *Proceedings of the 3rd ALEP user group workshop*. Saarbrücken, 1997

Badia, T., M. Pujol, A. Tuells, J. Vivaldi, L. de Yzaguirre & T. Cabré (1998) "IULA's LSP Multilingual Corpus: compilation and processing". Presented at the 1st ELRA Conference, Granada, 1998. URL: http://www.iula.upf.es/corpus/corpubca.htm

Badia, T. & À. Egea (2000) A strategy for the syntactic parsing of corpora: from Constraint Grammar output to unification-based processing. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, vol. II, Athens, 2000

Carmona, J., S. Cervell, L. Màrquez, M.A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé, J. Turmo (1998) An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *First International Conference on Language Resources and Evaluation (LREC'98)*. Granada, Spain, 1998

Fabra i Poch, P. (1956) *Gramàtica catalana (amb Prefaci de Joan Coromines)*. Barcelona: Teide.

DIEC (1996) *Diccionari de la Llengua Catalana.* Institut d'Estudis Catalans.

DLC (1995) *Diccionari de la Llengua Catalana.* Enciclopèdia Catalana.

Karlsson, F. *et. al.* (1995) *Constraint Grammar: A Language-Independent Formalism for Parsing Unrestricted Text.* Mouton de Gruyter: Berlin/New York.

Morel, J. *et. al.* (1997) El corpus de l'IULA: etiquetaris. IULA Papers: Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra, Barcelona. Sèrie Informes, 18. (2nd edition: revised and amended).

Rigau, G. (1998) *Automatic acquisition of lexical knowledge from MRDs* PhD dissertation, presented at the Universitat Politècnica de Catalunya, Barcelona.

Simpkins (1995) *Linguistic Development and Processing. ALEP-2.* European Comission.

Tapanainen, P. (1996) *The Constraint Grammar Parser CG-2.* Department of General Linguistics, University of Helsinki, Helsinki. Publications, number 27.

Tuells, T. (1998) "Constructing and Updating the Lexicon of a Two-Level Morphological Analyzer from a Machine-Readable Dictionary". In *Proceedings of the First International Conference on Language Resources & Evaluation*, Granada 1998.

[5] Unfortunately, we are still unable to publish accuracy and speed data of our system.