XML rules for enclitic segmentation *

Fco. Mario Barcala, Miguel A. Molinero, and Eva Domínguez

Centro Ramón Piñeiro, Ctra. Santiago-Noia km. 3, A Barcia,
15896 Santiago de Compostela, Spain
{fbarcala,molinero,edomin}@cirp.es

1 Introduction

Sentence word segmentation is an important task in robust part-of-speech (POS) tagging systems. In some cases this is relatively simple, since each textual word (or token) corresponds to one linguistic component. However, there are many others where segmentation can be very hard, such as those of contractions, verbal forms with enclitic pronouns, etc., where the same token contains information about two or more linguistic components. There are two main approaches to solving these difficult cases:

1. To treat tokens as a whole, and to extend the tagset to represent these phenomena. For instance, the Galician word `colléullelo` (*he or she caught it from them*) could be tagged as `Vei3s+Rad3ap+Raa3ms`, that is, verb (`V`), past simple (`ei`), 3rd person (`3`) singular (`s`) with an atonic pronoun (`Ra`), dative (`d`), 3rd person (`3`) masculine or feminine (`a`), plural (`p`) and another atonic pronoun (`Ra`), accusative (`a`), 3rd person (`3`), masculine (`m`), singular (`s`). Another widely alternative would be to tag `colléullelo` as `Vei3s2`, where the last `2` indicates that the verb has two enclitic pronouns.
2. To segment compound tokens, separating the components. For example, the same word can be broken into three parts: `colléu`, `lle` and `lo`. In this case, the components could be tagged separately as `Vei3s`, `Rad3ap` and `Raa3ms` respectively.

Although EAGLES [1] points towards using the second approach when these phenomena occur frequently, most papers and systems are based on the first approach [2] [3]. It is the simplest one, because there is no need to change current taggers [4], and it performs well with languages that hardly present these cases, for example English or French, but it presents several problems with others which have many occurrences of several linguistic components within the same word.

Such is the case of Galician, and to a lesser extent, other romance languages such as Spanish. In these cases, using the first approach, the tagset size would be greatly increased, and sometimes the POS tagging tasks would be impractical

* This paper results from a project developed at Centro Ramón Piñeiro para a Investigación en Humanidades and is partially supported by Ministerio de Educación y Ciencia (TIN2004-07246-C03-01), Xunta de Galicia (PGIDIT05PXIC30501PN) and Universidade de Vigo.

due to the need for an extremely large training corpus in order to obtain an acceptable tagger behaviour [5]. Therefore, works which choose the first option do not explain how to solve other POS related problems: how to assign lemmas to compound tokens, how to use the system inside a more complex one (for instance, a translation machine or a parser), etc. which in other works concerning the second alternative are trivial processes.

In this paper we follow guidelines indicated by EAGLES, and we explain the internals of a highly configurable system which segments compound tokens into its components. It must be used in combination with a tagger which solves segmentation ambiguities [6] [7] so as to perform jointly all POS tagging related tasks.

We will center our attention on the hardest item in segmentation, that is, the processing of verbal forms with enclitic pronouns. Contrary to other ad-hoc systems, to do this we use easily configurable high level XML [8] rules outside the system code, which will allow linguists to adapt the system to their particular needs and languages. We have applied it to the Galician language [9], but the system is generic enough to be applied to most romance languages and any tagset.

2 Main system

Our system has to decide if a word is a verb with enclitic pronouns, and, if so, has to segment its different linguistic components and pretag them (with all their valid tags) to help the tagger to decide which tags are the correct ones. To do so, it needs:

1. A lexicon containing the greatest possible number of verbal forms with their tags and lemmas.
2. A lexicon containing the greatest possible number of verbal stems capable of presenting enclitic pronouns. It must include the stem, the different tags which it can take, and the lemma of each entry.
3. A lexicon with all the valid combinations of enclitic pronouns (the character sequences only).
4. A lexicon with all enclitic pronouns with their possible tags and lemmas.
5. A set of morphosyntactic rules which guide the segmentation.

It takes a long time to build these lexicons, particularly the first two, but their building process is simple and there are techniques to use them without computational problems [10].

The main system is responsible for performing the segmentation and calls the following two subsystems to do all its tasks:

– The verbal subsystem, which determines if segmentations proposed by the main system are correct. If so, it assigns tags to the verbal part, filtering out impossible ones.

– The enclitic pronoun subsystem, which assigns tags to the enclitic pronouns and filters out inappropriate ones.

In a more detailed way, first of all, the main system finds candidate words for a verb with enclitic pronouns. It processes the text word by word and calls the verbal subsystem only when it finds a word in which:

– The final or enclitic pronoun part is a valid combination of one or more enclitic pronouns, that is, it is in the third lexicon listed above.
– The initial or verbal part is in the lexicon of verbal stems which can present enclitic pronouns.

It then passes the possible segmentation to the verbal subsystem: the verbal part, and the enclitic pronoun part. Note that it could call the verbal subsystem several times for the same word, since there could be different possible segmentations. After this, it uses the enclitic pronouns lexicon to segment the enclitics and to call the enclitic pronouns subsystem for each one.

3 Verbal subsystem

The verbal subsystem receives the verb part and the enclitic pronoun part from the main system and decides, by checking against its rules, if it is a valid segmentation. If so, it assigns candidate POS tags to the different linguistic components.

To avoid building an ad-hoc system, we externalize these advanced if-then rules through an XML file.

3.1 Verbal rules description

In figure 1 we show the XML DTD of this file. As can be seen, a rules document is a set of one or more rules, and each rule has one or more conditions (if) with their action (then).

A condition part has the following elements:

– **target**: the target of the evaluation of the rule. It can be *verb_part*, that is, the condition will be applied to the verbal part, *enclitic_part*, the condition will be applied to the enclitic pronoun part, *complete_form*, the condition will be applied to the complete form (the verb part and the enclitic pronoun part joined together) and *verb_tag*, the condition will be applied to the set of tags of the verbal part.
– **content**: the condition itself. It contains one or several *evaluation* elements with the evaluation expressions. The *evaluation* element has the *at* attribute to determine which portion of the target has to be matched with the expression. Several evaluation elements are interpreted as a logical OR between them.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT document (rule+,default_rule?)>
<!ELEMENT rule (condition+)>
<!ELEMENT default_rule (condition+)>
<!ELEMENT condition (target,content,action,check_default,filter*)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT content (evaluation+)>
<!ELEMENT action (#PCDATA)>
<!ELEMENT check_default (#PCDATA)>
<!ELEMENT filter (#PCDATA)>
<!ELEMENT evaluation (#PCDATA)>
<!ATTLIST evaluation at (all|end|begin) #REQUIRED>
```

Fig. 1. Verb rules DTD.

- **action**: the action to be executed if the evaluation condition is true. It can be *continue*, that is, to continue evaluating the next condition, *accept*, which finishes the evaluation confirming that it is indeed a correct segmentation, *reject*, which finishes the evaluation confirming that it is not a correct transformation, and *filter*, which is the same as *continue* but removes the tags specified in *filter* elements.
- **check_default**: Determines if *default_rule* must be evaluated.
- **filter**: It specifies a set of tags that must be removed before continuing if the *action* element contains *filter*. It contains a boolean expression that is checked against all the tags of *verb_part*. Tags which match the boolean expression are removed and the system continues with its comparisons. This is useful when there are non-valid tags for current segmentation.

The subsystem tests the rules from top to bottom. If an *evaluation* expression of a *condition* matches the specified portion of the *target*, the corresponding *action* is executed. The process stops when there is an *accept* or a *reject* in an executed action or when all rules have been tested. If there is not a matching rule which rejects the segmentation, the default behaviour is to accept it.

4 Enclitic pronoun subsystem

The enclitic pronoun subsystem is called by the main system once for each enclitic pronoun. It receives the *verb_part*, the *enclitic_part* and the enclitic to be processed, and assigns possible POS tags for the enclitic being processed. It also works with XML rules similar to the ones above.

4.1 Enclitic pronoun rules

Pronoun rules could be developed in the same way as verb ones, but we have decided to add some modifications in order to make the design of these rules

easier. As we can see in the DTD in figure 2, the main differences with respect to verbal rules are:

1. There is no *default_rule*.
2. Rules have a *type*, which can be *last_enclitic*, that is, the rule is applied if the received enclitic is the last one of the *enclitic_part*, or *intermediate_enclitic*, where the rule is applied if the received enclitic is not the last one.
3. Inside each rule there is an *enclitic* element which contains the enclitic to which this rule will be applied.
4. The *evaluation* element has three additional possibilities: *first_enclitic*, *next_enclitic* and *last_enclitic*, to specify matchings towards the first or the last enclitic of the *enclitic_part* or to the *next_enclitic* with respect to which it is being evaluated.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT document (rule+)>
<!ELEMENT rule (enclitic, condition+)>
<!ATTLIST rule type (intermediate_enclitic|last_enclitic) #REQUIRED>
<!ELEMENT condition (target,content,action,filter*)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT content (evaluation+)>
<!ELEMENT action (#PCDATA)>
<!ELEMENT filter (#PCDATA)>
<!ELEMENT evaluation (#PCDATA)>
<!ATTLIST evaluation at (all|end|begin|first_enclitic|
                         next_enclitic|last_enclitic) #REQUIRED>
```

Fig. 2. Enclitic pronoun rules DTD.

So, an *action* is executed if the enclitic to be processed matches the *enclitic* element of a rule, its position satisfies *type* attribute, and *evaluation* elements are verified.

5 Complete example

Let's suppose that the verbal rules file includes only the rule shown in figure 3, and the enclitic pronoun rules file includes only that in figure 4. Let's suppose also that `colléullelo` appears in the processing text. The main system analyzes it and concludes that it can be segmented into `colléu`, the *verb_part* and `llelo`, the *enclitic_part*. It reaches this conclusion because `colléu` is in the verb stems lexicon, and `llelo` is in the valid enclitic combinations lexicon.

The main system calls the verb subsystem, and it applies its rules. The rule in figure 3 says that if the verb part ends with `ei`, `éi`, `eu`, `éu`, etc., and the enclitic pronoun part starts with `o`, `lo`, etc., then it is not a valid segmentation.

```
<rule>
  <condition>
    <target>verb_part</target>
    <content>
      <evaluation at="end">
        ei OR éi OR eu OR éu OR ou OR óu OR iu OR íu OR ai OR ái
      </evaluation>
    </content>
    <action>continue</action>
    <check_default>no</check_default>
  </condition>
  <condition>
    <target>enclitic_form</target>
    <content>
      <evaluation at="begin">
        o OR lo OR a OR la OR los OR las OR -lo OR -la OR -los OR -las
      </evaluation>
    </content>
    <action>reject</action>
    <check_default>no</check_default>
  </condition>
</rule>
```

Fig. 3. Verbs rule example.

In our case, `colléu` ends with `éu`, but the enclitic pronoun part does not start with the combinations contained in the second condition evaluation, so `colléu` as *verb_part* and `llelo` as *enclitic_part* is a valid segmentation (because there is no rule which says otherwise)[1]. As there is no *filter* element, the verb subsystem assigns all tags present in the verb stems lexicon (in this case only `Vei3s0` tag).

Finally, the main system calls the enclitic pronoun with each enclitic, first with `lle` and then with `lo`. The rule in figure 4 says that if `lle` appears in the middle of the enclitic part, if the following enclitic is `lo`, `la`, etc. then, tag `Rad3as` must be filtered (removed). That is, although in the enclitic pronouns lexicon `lle` appears with the two possible tags (singular, `Rad3as` and plural, `Rad3ap`), `llelo` is always *it from them*, and never *it from him or her*, so the singular tag must be removed.

So, the system concludes that `colléullelo` must be segmented into `colléu`, as `Vei3s0`, `lle`, as `Rad3ap` and `lo`, as `Raa3ms`

6 Additional issues

For the purpose of clarity we have only described the basic functionality of the system, but it has many other possibilities:

[1] `colléulo`, `colléua`, etc. would not generate valid segmentation alternatives, even though `lo` and `a` are valid enclitic combinations.

```
<rule type="intermediate_enclitic">
  <enclitic>lle</enclitic>
  <condition>
    <target>enclitic_part</target>
    <content>
      <evaluation at="next_enclitic">
        lo OR la OR los OR las OR -lo OR -la OR -los OR -las
      </evaluation>
    </content>
    <action>filter</action>
    <filter>Rad3as</filter>
  </condition>
  <condition>
    <target>enclitic_form</target>
    <content>
      <evaluation at="next_enclitic">
        NOT lo AND NOT la AND NOT los AND NOT las AND NOT -lo
        AND NOT -la AND NOT -los AND NOT -las
      </evaluation>
    </content>
    <action>filter</action>
    <filter>Rad3ap</filter>
  </condition>
</rule>
```

Fig. 4. Enclitic pronouns rule example

Grouping conditions

There is a *resolution* element which can appear after the first condition to group other conditions. It allows the grouping of several conditions affected by the first one, minimizing the number of rules that have to be created.

True rules

We can define rules without *evaluation* elements. These rules have no *target* or *content* and are always evaluated as true, their *action* element running in all cases, in the case of verbal rules, or if the enclitic to be processed satisfies *type* and *enclitic* elements, in the case of enclitic pronoun rules.

Wildcards

Some rule elements allow the use of the ? wildcard, which matches any character. This is the case of *filter* and *evaluation* elements.

External functions

filter element has an optional parameter, *function*. If present, it allows external functions to be specified to perform different kinds of transformation. The parameters of the functions are specified in a *param* attribute of the *filter* element. This issue is very helpful when carrying out specific treatments.

Word reconstruction

So far, the system can split a verbal form with enclitic pronouns into the corresponding parts and assign possible tags to them. The final problem to solve is that the result chunks are not linguistically consistent. For instance, *colléu* is not a valid word when it appears alone, it must be *colleu*, without accent, tag *Rad3ap* refers to the isolated word `lles` and not to `lle`, and `lo` is an alomorph of `o`. So, the main system has to undo morphological changes which take place when the verbal part and the enclitic part are joined.

To do this for the enclitic pronouns, we use the external built-in function *replace* in the *filter* element of enclitic rules, which makes the replacement. *filter* rule element of figure 4 must be `<filter function=‘‘yes’’ param=‘‘lles’’>replace</filter>`, and the case of `lo` is solved by adding a simple true rule since in Galician `lo` is always, and only, an alomorph of `o`.

For the verbal part, the system makes use of lexicon lemmas. Once the verbal form has been segmented as we explained earlier, the system reconstructs the original form of the verb part using tags and lemmas of the resulting item to access these lexicons and to obtain the original form.

For instance, starting with the system result of the example in the previous section, first of all it obtains the lemma of `colléu`, looking into the verbal stems lexicon. Then, it searches tag `Vei3s0` and lemma `coller` in the verbal forms lexicon, obtaining the `colleu` form (without accent).

So, the final output of the system is `colleu`, as `Vei3s0`, `lles`, as `Rad3ap` and `o`, as `Raa3ms`

7 Conclusions

Sentence word segmentation is a very complex and important task in almost all natural language processing applications. Several works conceal or obviate the difficulties evolved in this process. In some cases, they adopt an easy partial solution acceptable for certain languages and applications, and, in others, they rely on a later or previous phase for solving it. However, there are hardly any papers with explanations describing how this later or previous phases have to be done.

In this paper we have described these problems, focusing on part-of-speech tagging tasks, and propose a solution for one of them: the segmentation of verbal forms which contain enclitic pronouns. We have presented a generic verb processing system, which segments and pretags verbs which have enclitic pronouns joined to them.

As we have seen, the system does not limit its function to segmentation, since it pretags the different linguistic components of a verbal form with enclitics, and removes invalid tags for its context. This innovative issue will be useful for part-of-speech taggers, which can use this information to avoid making certain errors, thus improving its results.

Although we have applied it to the Galician language, it can be easily adapted to other romance languages. The generic rule system we have designed allows rules to be written on the basis of XML files. This, combined with the use of lexicons, makes this adaptation simple and independent of the system internals.

References

1. Expert Advisory Group on Language Engineering Standards (EAGLES). Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A Common Proposal and Applications to European Languages. In *EAGLES Document EAG-CLWG-MORPHSYN/R*, May, 1996.
2. José L. Aguirre Moreno, Alberto Álvarez Lugrís, Xavier Gómez Guinovart. Aplicación do etiquetario morfosintáctico do SLI ó corpus de traducción TECTRA. In *Viceversa*, pages 207-231, 2002.
3. Carreras, X., I. Chao, L. Padró, M. Padró. FreeLing: An Open-Source Suite of Language Analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (LREC'04), Lisbon, Portugal, 2004.
4. Brants, T. A statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000)*, Seatle, 2000.
5. Jorge Graña. Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural. In *Doctoral thesis, Universidad de La Coruña*, Spain, 2000.
6. Jorge Graña, Miguel A. Alonso, Manuel Vilares. A Common Solution for Tokenization and Part-of-Speech Tagging: One-Pass Viterbi Algorithm vs. Iterative Approaches. In Petr Sojka, Ivan Kopecek and Karel Pala (eds.), *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Artificial Intelligence*, pp. 3-10, Springer-Verlag, Berlin-Heidelberg-New York, 2002.
7. Jorge Graña, Fco. Mario Barcala, Jesús Vilares. Formal Methods of Tokenization for Part-of-Speech Tagging. In *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pp. 240-249, Springer-Verlag, Berlin-Heidelberg-New York, 2002.
8. World Wide Web Consortium. http://www.w3c.org
9. Rosario Álvarez, Xosé Xove. Gramática da Lingua Galega. In *Editorial Galaxia*, Vigo, Spain, 2002.
10. Jorge Graña, Fco. Mario Barcala, Miguel A. Alonso, Compilation Methods of Minimal Acyclic Finite-State Automata for Large Dictionaries. In Bruce W. Watson and Derick Wood (eds.), *Implementation and Application of Automata*, volume 2494 of *Lecture Notes in Computer Science*, pp. 135-148, Springer-Verlag, Berlin-Heidelberg-New York, 2002.