TALN 2003, Batz-sur-Mer, 11–14 juin 2003

Extracting XML syntactic chunks from Portuguese corpora

Caroline Gasperin(1), Renata Vieira(1), Rodrigo Goulart(1), Paulo Quaresma(2)
(1) PIPCA - Unisinos
São Leopoldo, Brazil
{caroline,renata,rodrigo}@exatas.unisinos.br
(2) UEVORA
Evora, Portugal
pq@di.uevora.pt

Abstract

The Portuguese language has a great number of speakers distributed on Europe, South America, Asia and Africa but research and development on Portuguese processing are still limited compared to languages such as English, French and Spanish. Regarding parsing tools, one basic component for NLP systems, the CURUPIRA parser (Martins, 2002) and the PALAVRAS parser (Bick, 2000) have been recently made available. The PALAVRAS parser is a robust tool and we have used it as a basis in the development of previous work on Portuguese NLP applications. As a way to promote the applicability of this tool we propose an XML encoding for its output. According to our proposal, linguistic information may be provided in XML and it can be tailored to the needs of different NLP application. The paper illustrates the use of our tool and schemes by describing two applications that make use of different linguistic information extracted from Portuguese parsed corpus.

1 Introduction

The Portuguese language has a great number of speakers but the availability of computational tools is limited, specially, regarding parsing tools. (A list of existing tools for Portuguese can be found at http://www.linguateca.pt/ferramentas.html.) We have the CURUPIRA parser available since 2002 (Martins, 2002), and the PALAVRAS parser since 2000 (Bick, 2000). PALAVRAS is a parser based on constraint-grammar formalism developed at the Institute of Language and Communication of the University of Southern Denmark. It is available on the Internet[1] and is a robust tool. Still, one problem we find when using the analyses provided by PALAVRAS is that it is not in a standard format, so the extraction of syntactic information from parsed corpora depends on specific tools that have to be built for each intended application.

In this paper we present our tool and schemes for XML encoding the output of PALAVRAS. The parser has been used as a basis for previous work related to Portuguese processing (Vieira et al., 2000; Gasperin *et al.*, 2001). In these works, different tools for extracting syntactic

[1] http://visl.hum.sdu.dk/visl/pt

```
STA:fcl
SUBJ:np
=>N:num('três' M P <card>) Três
=H:n('acidente' M P) acidentes
=N<:adj('grave' M P) graves
P:v-fin('marcar' PS/MQP 3P IND) marcaram
ACC:np
=>N:art('o' <artd> M S) o
=H:n('fim_de_semana' M S) fim_de_semana
.
```

Figure 1: Parsed sentence.

information had to be developed, adding time costs to the projects. In order to make the use of syntactic information from parsed corpora a simpler task, we developed a tool that generates the XML encoding for the PALAVRAS output. Our tool can also extract specific chunks from the parsed corpora according to the linguistic information needed for different NLP tasks. In this paper we illustrate the extraction of chunks for different applications: we use lists of NPs for anaphora resolution and triples of subject-verb-object to acquire knowledge from texts.

The paper is organized as follows. In Section 2 we present the parser output format. Section 3 presents our XML encoding principles. Section 4 presents how we generate XML output and extract chunks from the parsed corpus. Section 5 shows the use of our tool in different NLP applications. Our concluding remarks are presented on section 6.

2 PALAVRAS output

Take the sentence in Portuguese "Três acidentes graves marcaram o fim de semana." (Three serious accidents marked the weekend.). The parser output for this sentence is shown on Figure 1.

On each line of the figure, the first symbol represents the syntactic function ('SUBJ'=subject, 'N'=noun modifier, 'H'=head, 'P'=predicator, 'ACC'=direct object); after the ':' there is the syntactic form for groups of words and POS-tags for single words ('np'=noun phrase, 'n'=noun, 'v'=verb, etc.); in brackets there is the word canonical form and other inflectional tags; after the brackets there is the word as it occurs in the corpus. The '=' signs in the beginning of each line represent the level of the phrase in the parsing tree.[2] Because this is not a standard format, the extraction of syntactic information from analysed corpora requires parsing it for different NLP applications. To simplify the use of parsed corpora, we transform PALAVRAS output into XML chunks. In this way we can use the XML tools already available to access the information that is needed.

3 Encoding principles

Our proposal is mainly influenced by MMAX (Müller & Strube, 2001b), the annotation tool that we have been using for several experiments on corpus annotation, as reported in (Vieira *et al.*, 2002b; Salmon-Alt & Vieira, 2002; Vieira *et al.*, 2002a). In (Müller & Strube, 2001a)

[2] A complete description of the tagset symbols is available at http://visl.hum.sdu.dk/visl/pt/info/symbolset-manual.html.

```
<!ELEMENT words(word*)>
<!ELEMENT word(#PCDATA)>
<!ATTLIST word
  id ID #REQUIRED
>
```

(a)

```
<!ELEMENT text (paragraph+)>
<!ELEMENT paragraph (sentence*)>
<!ATTLIST paragraph
  id ID #REQUIRED
>
<!ELEMENT sentence (EMPTY)>
<!ATTLIST sentence
  id ID #REQUIRED
  span CDATA #REQUIRED
>
```

(b)

```
<!ELEMENT markables (markable*)>
<!ELEMENT markable (#PCDATA)>
<!ATTLIST markable
  id ID #REQUIRED
  span CDATA #REQUIRED
  type CDATA #REQUIRED
  member CDATA #IMPLIED
  pointer IDREF #IMPLIED
>
```

(c)

Figure 2: MMAX DTDs.

the following encoding architecture is presented. There is a base input file that describes the corpus tokens codified as <word> elements, according to the DTD presented in Figure 2(a). A second input file identifies the text structure (paragraphs and sentences), according to Figure 2(b). The output file contains the annotation done over the corpus. The annotation is codified by <markable> elements according to the DTD shown in Figure 2(c). Other attributes can be specified by the user according to his own annotation task.

Since we intend to follow standards for corpora annotation (Ide & Romary, 2002), we adopted the words file as proposed by (Müller & Strube, 2001a) as our basic file to which every other linguistic information should refer to. The words file for our example is shown on Figure 3.

In our scheme we identify syntactic structures as <chunk> elements into the chunks file (whose DTD is an extended version of the text structure DTD) and additional POS information is described in a POS file, both referring to the basic words file. Next section presents these files in detail.

4 Extracting chunks from parsed corpora

The program that transforms the output of the parser into XML, first generates Prolog terms corresponding to each parsed sentence. Figure 6 shows the terms for the parsed sentence in Figure 1.

From the Prolog terms the following files are generated: a basic words file, a POS file, and the chunks file which is tailored according to the application needs. We can indicate the kind of chunks to be extracted, just informing it as parameters of a Prolog predicate. For example, if we intend to extract noun phrases, we need to inform the program the value “np” as parameter.

The XML chunks are specified according to the DTD shown on Figure 4. The syntactic function of a chunk is given by its *function* attribute. The attribute *form* corresponds to the syntactic form of a word group or to the POS category of a single word. The chunks *span* attribute refers to <word> elements of the basic file. Figure 5 shows the complete chunks file for our example

```
<words>
  <word id="word_1">Três</word>
  <word id="word_2">acidentes</word>
  <word id="word_3">graves</word>
  <word id="word_4">marcaram</word>
  <word id="word_5">o</word>
  <word id="word_6">fim_de_semana</word>
  <word id="word_7">.</word>
</words>
```

Figure 3: Words file.

```
<!ELEMENT text (paragraph+)>
<!ELEMENT paragraph (sentence*)>
<!ATTLIST paragraph
  id ID #REQUIRED
>
<!ELEMENT sentence (chunk*)>
<!ATTLIST sentence
  id ID #REQUIRED
  span CDATA #REQUIRED
>
<!ELEMENT chunk (chunk*)>
<!ATTLIST chunk
  id ID #REQUIRED
  function CDATA #REQUIRED
  form CDATA #REQUIRED
  span CDATA #REQUIRED
>
```

Figure 4: Chunks DTD.

```
<text>
  <paragraph id="paragraph_1">
    <sentence id="sentence_1" span="word_1..word_14">
      <chunk id="chunk_1" function="subj" form="np" span="word_1..word_3">
        <chunk id="chunk_2" function="n" form="num" span="word_1"/>
        <chunk id="chunk_3" function="h" form="n" span="word_2"/>
        <chunk id="chunk_4" function="n" form="adj" span="word_3"/>
      </chunk>
      <chunk id="chunk_5" function="p" form="v_fin" span="word_4"/>
      <chunk id="chunk_6" function="acc" form="np" span="word_5..word_6">
        <chunk id="chunk_7" function="n" form="art" span="word_5"/>
        <chunk id="chunk_8" function="h" form="n" span="word_6"/>
      </chunk>
    </sentence>
  </paragraph>
</text>
```

Figure 5: Complete chunks file.

```
sentence(syn(
sta(fcl),
subj(np,
n(num('três','M','P','<card>'),'Três'),
h(n('acidente','M','P'),'acidentes'),
n(adj('grave','M','P'),'graves')),
p(v_fin('marcar','PS/MQP','3P','IND'),'marcaram'),
acc(np,
n(art('o','M','S'),'o'),
h(n('fim_de_semana','M','S'),'fim_de_semana', '.')))).
```

Figure 6: Prolog terms.

```
<!ELEMENT words (word*)>

<!ELEMENT word (n|prop|adj|v|art|pron|adv|num|
  prp|intj|conj)>
<!ATTLIST word id ID #REQUIRED>

<!ELEMENT n (secondary_n?)>
<!ATTLIST n
  canon CDATA #REQUIRED
  gender (M | F) #REQUIRED
  number (P | S) #REQUIRED
>

<!ELEMENT prop (secondary_prop?)>
<!ATTLIST prop
  canon CDATA #REQUIRED
  gender (M | F) #REQUIRED
  number (P | S) #REQUIRED
>

<!ELEMENT adj (secondary_adj?)>
<!ATTLIST adj
  canon CDATA #REQUIRED
  gender (M | F) #REQUIRED
  number (P | S) #REQUIRED
>

<!ELEMENT v ((fin|inf|pcp|ger), sec-
ondary_v?)>
<!ATTLIST v canon CDATA #REQUIRED>
<!ELEMENT fin EMPTY>
<!ATTLIST fin
  person (1S|2S|3S|1P|2P|3P) #RE-
QUIRED
  tense (PR|IMPF|PS|FUT|IMP) #RE-
QUIRED
  mode (IND|SUBJ) #REQUIRED
>
<!ELEMENT inf EMPTY>
<!ELEMENT pcp EMPTY>
<!ATTLIST pcp
  gender (M|F) #REQUIRED
  number (P|S) #REQUIRED
>
<!ELEMENT ger EMPTY>
...
```

Figure 7: Words POS DTD.

```
<words>
  <word id="word_1">
    <num canon="três" gen-
der="M" number="P">
      <secondary_num tag="card"/>
    </num>
  </word>
  <word id="word_2">
    <n canon="acidente" gen-
der="M" number="P"/>
  </word>
  <word id="word_3">
    <adj canon="grave" gen-
der="M" number="P"/>
  </word>
  <word id="word_4">
    <v canon="marcar">
      <fin tense="PS/MQP" per-
son="3P" mode="IND"/>
    </v>
  </word>
  <word id="word_5">
    <art canon="o" gender="M" num-
ber="S">
      <secondary_art tag="artd"/>
    </art>
  </word>
  <word id="word_6">
    <n canon="fim_de_semana" gen-
der="M" number="S"/>
  </word>
</words>
```

Figure 8: Words POS file.

sentence.

The POS file is specified according to the DTD shown on Figure 7. This DTD was based on the PALAVRAS tag set. For our example sentence, we have the POS file shown on Figure 8.

5 Using XML chunks

In this section we illustrate briefly how we use Portuguese syntactic chunks in two different applications: anaphora resolution and knowledge extraction from texts. Both applications are at early stages of development, our intention in this section is mainly to show how the chunks can serve different purposes, rather than discuss results related to these applications. We believe that the examples may help other users interested in using the tool for other applications.

```
<paragraph "paragraph_1">
  <sentence id="sentence_1" span="word_1..word_14">
    <chunk id="chunk_1" function="subj" form="np" span="word_1..word_3">
      <chunk id="chunk_2" function="h" form="n" span="word_2"/>
    </chunk>
...
  </sentence>
</paragraph>
```

Figure 9: NP chunks

5.1 Anaphora and coreference resolution

We are developing a multi-lingual tool for anaphora resolution of definite descriptions (*...the boys...*), demonstrative noun phrases (*...these boys...*) and pronouns (*...they...*). The main goal is to identify the antecedents for these anaphoric expressions. As we are considering those cases where the antecedent is a noun phrase, we need to extract all NPs from the parsed corpus. For extracting NP chunks, we select chunks whose syntactic form is "np". Figure 9 shows NP chunks for our example sentence.

From NP chunks, we select anaphors and antecedent candidates. Our anaphors are identified by the presence of definite article, demonstrative and personal pronouns. This information is given in the POS file. Then, we apply heuristics to identify the correct antecedent among the candidates. The heuristics to be used are based on previous studies about resolution of nominal referring expressions (Vieira & Poesio, 2000; Lappin & Leass, 1994; Strube *et al.*, 2002) and they are not discussed here. These tasks are performed by a set of stylesheets. Each one is connected to another through pipes and it filters the information flowing through the system (Gamma *et al.*, 1995). There are three main steps: anaphor selection, candidates selection and markables generation (Figure 10 A, B and C respectively). Each step corresponds to one stylesheet.

The Anaphor selection task (A) receives chunks and uses POS information to select the anaphors. In the selection a node <anaphor> is created, it contains the *span* attribute referring to the words file, and another node <header> contains the head noun of the NP (Figure 11(a)).

The Candidates selection task (B) selects antecedent candidate from the NP chunks (as shown in Figure 11(b)).

Finally, the Markables generation task (C) matches head nouns using the <anaphor> and <candidate> information through the application of a sequence of heuristics informed in the rule base. The output is MMAX compatible, so the results can be visualized using the MMAX tool. Figure 12 shows the output markables.

5.2 Knowledge acquisition from texts

Our second application is related to experiments towards semi-automatic generation of conceptual maps from a parsed corpus. From parsed texts, we first extract triples of subject-verb-object. Over these triples we apply a set of filtering heuristics based on the frequency of the relations and frequency of the terms. We are also investigating whether different terms appearing in the same relations of other terms form a set of semantically related words. For extracting subject,

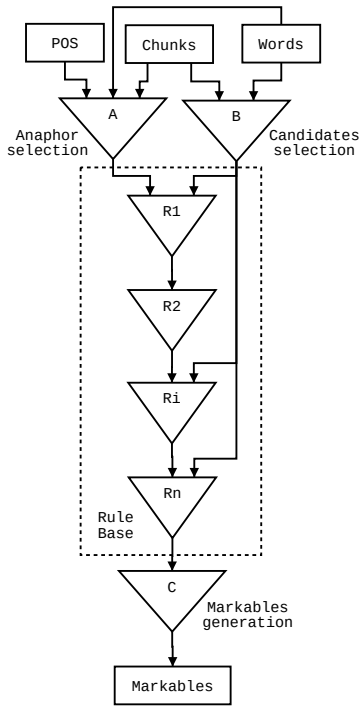

Figure 10: Anaphora resolution design

```
<anaphor span="word_5..word_8">
  <header>final</header>
</anaphor>
```

(a)

```
<candidate span="word_1..word_3">
  <header>acidentes</header>
</candidate>
<candidate span="word_5..word_6">
  <header>final</header>
</candidate>
```

(b)

Figure 11: Anaphor nodes (a) and Candidate nodes (b)

```
<markables>
  <markable id="markable_1" pointer="" span="word_5..word_8" classifica-
tion="discourse_new"/>
</markables>
```

Figure 12: Markable nodes

```
<paragraph id="paragraph_1">
  <sentence id="sentence_1" span="word_1..word_9">
    <chunk id="chunk_1" function="subj" form="np" span="word_1..word_3">
      <chunk id="chunk_2" function="h" form="n" span="word_2"/>
    </chunk>
    <chunk id="chunk_3" function="p" form="v_fin" span="word_4"/>
    <chunk id="chunk_4" function="acc" form="np" span="word_5..word_8">
      <chunk id="chunk_5" function="h" form="n" span="word_6"/>
    </chunk>
  </sentence>
</paragraph>
```

Figure 13: Subject and Object chunks

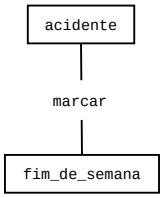

Figure 14: Example of a relation in a conceptual map.

verb and object chunks, we select that ones with *function* equal to “subj”, “acc”, and *form* equal “vp”, “v” and “np”.

For our example case, we consider (Figure 13) chunk_2 (subject head noun), chunk_3 (verb) and chunk_5 (object head noun), generating the triple “acidente-marcar-fim_de_semana”. Triples are extracted from chunks of the whole corpus; after filtering the set of triples relations between two concepts, a conceptual map is generated, as shown in Figure 14. We use the CMap tool (http://cmap.coginst.uwf.edu/) to generate the map. The heuristics for filtering the triples are being defined.

This methodology has been applied to a subset of the Portuguese Attorney General’s Office documents (Quaresma & Rodrigues, 2003). We have selected a subset of 40 documents having event descriptions. These documents were parsed and the correspondent XML chunks (subject, verb, object) were produced. The triples and the correspondent conceptual maps were created and are currently under analysis. These conceptual relations may be also used for the creation of an ontology of actions. In previous work (Saias & Quaresma, 2002) it is shown how to automatically transform conceptual relations into an ontology defined using the DAML+OIL/OWL semantic web language (DAM, 2000).

6 Concluding remarks

In this paper we presented a tool that extracts XML chunks from the PALAVRAS parser output. The chunks generated by our tool may reflect both the complete parsing and selected information tailored to a particular application. With the XML encoding of the linguistic information, different NLP applications can access it through already available XML tools.

Besides the advantages of using XML, our schemes are compatible with an existing annotation tool, MMAX. We have also presented two applications developed on the basis of the tool and

schemes presented here.

As current work we are adapting our encoding schemes to proposed standards (Ide & Romary, 2002). We are also developing a web interface to integrate our tools to the PALAVRAS parser. With this work we intend to promote the access and use of basic tools for the development of NLP applications for the Portuguese language.

Acknowledgments

We would like to thank CNPq(Brazil)/INRIA(France) and CAPES(Brazil)/FCT(Portugal) for their financial support. We are grateful to Eckhard Bick for his valuable help on the use of the parser PALAVRAS, Christoph Müller and Michael Strube for providing background for our annotation schemes, and Susanne Salmon-Alt.

References

(2000). *DAML+OIL – DARPA Agent Markup Language*. DAML, http://www.daml.org.

BICK E. (2000). *The Parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. PhD thesis, Århus University, Århus.

GAMMA E., HELM R., JOHNSON R. & VLISSIDES J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. New York: Addison-Wesley Publishing Company.

GASPERIN C., GAMALLO P., AGUSTINI A., LOPES G. & LIMA V. (2001). Using syntactic contexts for measuring word similarity. In *Proceedings of the Workshop on Semantic Knowledge Acquisition and Categorisation*, Helsink, Finland.

IDE N. & ROMARY L. (2002). Standards for language resources. In *Proceedings of the LREC 2002*, p. 839–844, Las Palmas de Gran Canaria.

LAPPIN S. & LEASS H. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, **20**(4).

MARTINS, R. T.; HASEGAWA R. N. M. (2002). *Curupira: um parser functional para o português*. Nilc-tr-02-06, USP-UNESP-UFSCAR, São Carlos.

MÜLLER C. & STRUBE M. (2001a). Annotating anaphoric and bridging expressions with MMAX. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue*, p. 90–95, Aalborg, Denmark.

MÜLLER C. & STRUBE M. (2001b). MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the IJCAI 2001*, p. 45–50, Seattle.

QUARESMA P. & RODRIGUES I. P. (2003). PGR: Portuguese attorney general's office decisions on the web. In BARTENSTEIN, GESKE, HANNEBAUER & YOSHIE, Eds., *Web-Knowledge Management and Decision Support*, Lecture Notes in Artificial Intelligence LNCS/LNAI: Springer-Verlag. To be published.

SAIAS J. & QUARESMA P. (2002). Semantic enrichment of a web legal information retrieval system. In T. BENCH-CAPON, Ed., *JURIX'2002 - Fifteenth Annual International Conference on Legal Knowledge and Information Systems*, London, UK: IOS Press.

SALMON-ALT S. & VIEIRA R. (2002). Nominal expressions in multilingual corpora: Definites and demonstratives. In *Proceedings of the LREC 2002*, Las Palmas de Gran Canaria.

STRUBE M., RAPP S. & MÜLLER C. (2002). The influence of minimum edit distance on reference resolution. In *Proceedings of the EMNLP 2002*, Philadelphia.

VIEIRA R. & POESIO M. (2000). An empirically-based system for processing definite descriptions. *Computational Linguistics*, **26**(4), 525–579.

VIEIRA R., SALMON-ALT S., GASPERIN C., SCHANG E. & OTHERO G. (2002a). Coreference and anaphoric relations of demonstrative noun phrases in multilingual corpus. In *Proceedings of the DAARC 2002*, Estoril.

VIEIRA R., SALMON-ALT S. & SCHANG E. (2002b). Multilingual corpora annotation for processing definite descriptions. In *Proceedings of the PorTAL 2002*, Faro.

VIEIRA ET AL. (2000). Extração de sintagmas nominais para o processamento de coreferência. In *Anais do V Encontro para o processamento computacional da Língua Portuguesa escrita e falada - PROPOR*, Atibaia.