The wraetlic NLP suite

Enrique Alfonseca[12], Antonio Moreno-Sandoval[3], José María Guirao[4] and Maria Ruiz-Casado[12]

[1]Computer Science Department, Universidad Autónoma de Madrid
{Enrique.Alfonseca,Maria.Ruiz}@uam.es
[2]Precision and Intelligence Laboratory, Tokyo Institute of Technology
[3]Laboratorio de Lingüística Informática, Universidad Autónoma de Madrid
antonio.msandoval@uam.es
[4]Software Engineering Department, Universidad de Granada
jmguirao@ugr.es

Abstract

In this paper, we describe the second release of a suite of language analysers, developed over the last five years, called *wraetlic*, which includes tools for several partial parsing tasks, both for English and Spanish. It has been successfully used in fields such as Information Extraction, thesaurus acquisition, Text Summarisation and Computer Assisted Assessment.

1. Introduction

Recently, there have appeared several freely available suites for performing the most basic operations in Natural Language Processing, typically tokenisation, sentence splitting, part-of-speech tagging, chunking and partial parsing. The availability of these resources facilitates largely the development of higher-level applications, such as semantic analysers, Information Extraction, dialogue interfaces and Question Answering systems, to cite but a few. There is already a number of Natural Language Processing (NLP) toolkits available, amongst which we may cite OAK (Sekine, 2002), NLTK (Loper and Bird, 2005), FreeLing (Carreras et al., 2004), GATE (Bontcheva et al., 2004), Ellogon (Petasis et al., 2002) and SProUT (Drozdzynski et al., 2005). Additionally, the OpenNLP (OpenNLP) initiative groups together several open-source projects for developing Natural Language Proceeding tools.

In this paper, we describe a suite of language analysis tools, mainly developed at the Universidad Autonoma de Madrid, called *wraetlic* (Alfonseca et al., 2005). These tools have been already applied to tasks as diverse Information Extraction, thesaurus acquisition (Alfonseca and Manandhar, 2002), Text Summarisation (Alfonseca et al., 2004) and Computer Assisted Assessment (Pérez et al., 2005).

The paper is structured as follows: Section 2 reviews other equivalent toolkits; next, Section 3 provides an overview of the wraetlic tools, its design, and describes the reason for the technical choices taken in the several modules. Finally, Section 4 describes the plans we have in mind for improving and extending the tools in the future.

2. Related work

A common classification (Cunningham et al., 1997; Petasis et al., 2002) groups existing systems into the following four types, according to the representation schema used for the linguistic annotations:

- **Additive or markup-based**, when the text is annotated with linguistic information with a markup scheme, such as SGML or XML. Existing toolkits in this category include LT-XML (McKelvie et al., 1997; Brew et al., 2000). As Petasis et al. (2002) points out, this approach has the advantage that a program may load from the XML document just the information wanted, resulting in small memory requirements, but they are usually criticised as the documents typically have to be re-parsed by each module because these systems are usually implemented as pipelines.

- **Referential or Annotation based**: in this case, the linguistic information consists of references to the textual data, which is kept separately. Also known as *stand-off annotation* (Thompson and McKelvie, 1997), it has been increasingly used to the point that it is considered a pre-requisite of the ISO/TC 37/SC 4 linguistic annotation framework (Ide and Romary, 2003). TIPSTER (Grishman, 1996), GATE (Bontcheva et al., 2004), Ellogon (Petasis et al., 2002) and JET (Grishman, 2005) belong to this category. It has as advantage the possibility of creating multiple annotations for a single document, possibly overlapping each other. GATE is probably the most widely used NLP suite at the moment, having been applied to more than 50 research projects.

- **Abstraction based**, when the original text is transformed into an internal data structure, that is theoretically grounded, as in the ALEP system (Simkins, 1994). SProUT (Drozdzynski et al., 2005) combines finite-state techniques and unification-based formalisms, and uses Typed Featured Structures as the data representation.

- **Systems without a uniform representation**, those that provide an architecture for communication and control, but do not impose a uniform data structure to be used by the different modules, such as the TalLab platform (Wolinski et al., 1998). The OAK (Sekine, 2002) modules also understand multiple formats (annotations, SGML...) when communicating with each other.

Cunningham et al. (2000) also provides a detailed analysis of the existing suites according to several features.

3. Wraetlic Overview

The wraetlic suite started to be developed in 2000 with the aim of having available an easy-to-use toolkit for processing English and Spanish, with a modular architecture that can be easily extended with new functionality and with alternative classes for new languages. Most of it has been implemented in Java and, although a few modules had to be implemented in C for the sake of efficiency or because of dependencies on external runtime libraries, alternate versions in Java are either already provided or being built at the moment. During the five years in which this suite has been developed, there have been just slight changes in the main design, even though many modules have been added and replaced over time. Therefore, we expect it will also be able to accommodate further improvements.

For simplicity, an additive annotation scheme based on XML was chosen. In this way, the system can be easily implemented as a pipeline of processes where the Operating System is responsible of communicating information from every module to the next one. To improve the efficiency, a Java API is also provided, so it is possible from a single Java program to call the different modules linearly, and in this way it is not necessary for every module to re-parse the XML documents. The occasional need to encode overlapping annotations or graph-structured information has been solved by using non-content XML nodes and hyperlinks inside the document, and a special module for retrieving the hyperlink targets as document portions. The use of XML also facilitates the construction of tools for visualising the results, given that a simple combination of XSL transformations with a web browser we can obtain the desired results with little effort.

In fact, we believe that the annotation schema proposed, being markup-based, is the main feature that distinguishes the wraetlic tools from most of the other toolkits that are now under active development[1].

Figure 1 shows a sample XML document containing one sentence after the parser has processed it. The toolkit currently includes all the following modules for English:

- Tokenisation and sentence splitting.
- Stemming.
- PoS tagging.
- Named Entity Recognition and Classification (NERC).
- Chunking and partial parsing.
- Word-Sense Disambiguation.
- Extract and headline generation.

Furthermore, they also include some of them for Spanish, including tokenisation, PoS tagging and chunking. The next subsections describe the technical details of these modules.

3.1. Segmentation and stemming

The tokeniser has been programmed as a list of regular expressions for defining the different tokens, such as words, numbers or punctuation symbols. Both flex and JFlex modules have been provided. For the sentence splitter, the pro-

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE document SYSTEM "yorkie.dtd">
<document>
  <header/>
  <body id="0" nextId="25586">
    <chapter id="25570">
      <header id="25567">
        <p id="26">
          <s id="27">
            <np det="none" person="3" number="singular" id="28">
              <w c="w" pos="NN" stem="CHAPTER" id="29">CHAPTER</w>
              <w c="w" abbreviation="yes" pos="NNP" stem="I" head="yes" id="30">I.</w>
            </np>
          </s>
        </p>
      </header>
      <section id="25562">
        <header id="25561">
          <p id="6281">
            <s id="6282">
              <np det="none" person="3" number="singular" id="6283">
                <w c="w" pos="NNP" stem="FERNANDO" id="6284">FERNANDO</w>
                <w c="w" pos="NNP" stem="NORONHA" head="yes" id="6285">NORONHA</w>
              </np>
              <w c="," pos="," id="6286">,</w>
              <advp entity="date" id="6287" calendarDate="////20/2/1832/0/t13/">
                <w c="w" pos="NNP" stem="FEBRUARY" id="6288">FEBRUARY</w>
                <w c="cd" pos="CD" id="6289">20</w>
                <w c="," pos="," id="6290">,</w>
                <w c="cd" pos="CD" id="6291">1832</w>
              </advp>
            </s>
          </p>
        </header>
        <body>
          <p id="6292">
            <s id="6293">
              <w c="w" pos="IN" id="6294">As</w>
              <w c="w" pos="RB" id="6295">far</w>
              <pp id="21236">
                <w c="w" pos="IN" id="6296" head="yes">as</w>
                <np number="singular" person="1" id="6297">
                  <w c="w" pos="PRP" head="yes" id="6298">I</w>
                </np>
              </pp>
              <vbar voice="passive" time="past" tense="finite" id="6299" args="+6302">
                <w c="w" pos="VBD" stem="be" head="yes" id="6300">was</w>
                <w c="w" pos="VBN" stem="enable" lexhead="yes" id="6301">enabled</w>
              </vbar>
              <vbar tense="infinitive" id="6302" subject="*6299" args="+6311">
                <w c="w" pos="TO" id="6303">to</w>
                <w c="w" pos="VB" stem="observe" head="yes" lexhead="yes" id="6304">observe</w>
              </vbar>
              <w c="," pos="," id="6305">,</w>
              <pp id="23660">
                <w c="w" pos="IN" id="6306" head="yes">during</w>
                <np det="definite" person="3" number="plural" id="6307">
                  <w c="w" pos="DT" id="6308">the</w>
                  <w c="w" pos="JJ" id="6309">few</w>
                  <w c="w" pos="NNS" stem="hour" head="yes" id="6310">hours</w>
                </np>
              </pp>
              <np number="plural" person="1" id="6311">
                <w c="w" pos="PRP" head="yes" id="6312">we</w>
              </np>
              <vbar time="past" tense="finite" id="6313" args="+6320">
                <w c="w" pos="VBD" stem="stay" lexhead="yes" head="yes" id="6314">stayed</w>
              </vbar>
              <pp id="23662">
                <w c="w" pos="IN" id="6315" head="yes">at</w>
                <np det="none" person="3" number="singular" id="6316">
                  <w c="w" pos="DT" id="6317">this</w>
                  <w c="w" pos="NN" stem="place" head="yes" id="6318">place</w>
                </np>
              </pp>
              <w c="," pos="," id="6319">,</w>
              <np det="definite" person="3" number="singular" id="6320">
                <w c="w" pos="DT" id="6321">the</w>
                <w c="w" pos="NN" stem="constitution" head="yes" id="6322">constitution</w>
              </np>
              <pp id="23664">
                <w c="w" pos="IN" id="6323" head="yes">of</w>
                <np det="definite" person="3" number="singular" id="6324">
                  <w c="w" pos="DT" id="6325">the</w>
                  <w c="w" pos="NN" stem="island" head="yes" id="6326">island</w>
                </np>
              </pp>
              <vbar time="present" tense="finite" id="6327">
                <w c="w" pos="VBZ" stem="be" lexhead="yes" head="yes" id="6328">is</w>
              </vbar>
              <w c="w" pos="JJ" id="6329">volcanic</w>
              <w c="," pos="," id="6330">,</w>
              <w c="w" pos="CC" id="6331">but</w>
              <w c="w" pos="RB" id="6332">probably</w>
              <w c="w" pos="RB" id="6333">not</w>
              <pp id="23666">
                <w c="w" pos="IN" id="6334" head="yes">of</w>
                <np det="indefinite" person="3" number="singular" id="6335">
                  <w c="w" pos="DT" id="6336">a</w>
                  <w c="w" pos="JJ" id="6337">recent</w>
                  <w c="w" pos="NN" stem="date" head="yes" id="6338">date</w>
                </np>
              </pp>
            </s>
          </p>
        </body>
      </section>
    </chapter>
  </body>
</document>
```

Figure 1: Sample sentence, extracted from *The Voyages of the Beagle*, with some syntactic annotation.

[1] LT-XML does not appear to be under development, as there have not been major releases in more than five years.

cedure chosen was the algorithm described by Mikheev (2002). It is, to our knowledge, the most accurate reported in related literature up to now, with an error rate between 0.28% and 0.45%.
The stemmer is also based on flex regular expressions, based in the LaSIE's open-source stemmer (Gaizauskas et al., 1995).

3.2. Part-of-speech tagger

For PoS tagging there are also several widely used algorithms, amongst which wraetlic provides the TnT procedure (Brants, 2000), which was the highest scoring algorithm reported when it was programmed (96.7%). Wraetlic uses the part-of-speech labels from the Penn Treebank (Marcus et al., 1993) for the English language, and the labels from the C-ORAL-ROM corpus (Moreno-Sandoval et al., 2005) for Spanish.

3.3. Named Entity Recognition and Classification

Named Entity Recognition has been divided into two separate steps: the temporal expressions are recognised with regular expressions, using a flex-generated program; whereas three different systems can be combined for identifying people, organisations, and locations: a Maximum Entropy classifier, Error-Driven Transformation List Learning, and automatically-learnt sure-fire rules (Mikheev et al., 1999; Alfonseca and Ruiz-Casado, 2005). The combination of systems is now a common procedure for NERC, and the combination with sure-fire rules attained the highest score in the last Message Understanding Conference (MUC-7) (SAIC, 1998).
In English, the NERC Maximum Entropy and Transformation List modules have been trained on the CoNLL-2003, the MUC-6 and the MUC-7 corpora. The complete module attains an F-score of 96.16% for people, 95.75% for locations and 91.71% for organisations on the CoNLL-2003 test corpus. In Spanish, the training corpus used is the CoNLL-2002.

3.4. Chunk parsers

For English, three different chunk parsers are provided. The most basic one is the **Noun Phrase (NP) chunker**, implemented as error-driven transformation lists, which was trained using the chunked section of the Penn Treebank as training corpus. The accuracy obtained is the same reported by Ramshaw and Marcus (1995), around 92%, but in our experiments we observed that many of the errors committed were due to random or systematic mistakes in both in the training and the test corpus. Rather than improving the learning algorithm, we focused on improving the quality of the training material. In this way, one person-month was spent performing a semi-automatic engineering work, classifying these errors in the training and test corpora, and correcting them by hand, expecting that the learner would find it easier to learn from a cleaner corpus. This led to a substantial increase in the accuracy of the transformation list learnt.
A further improvement is obtained if the Noun Phrase chunker is combined with a **Quantifier Phrase chunker**, that learns multiword quantifiers such as *at least 100*, *hundreds of*, *between 3 and 5*, etc. This chunker was found to be useful if used before the Noun Phrase chunker, because these quantifiers are easy to recognise, but they were the cause of many of the errors while chunking NPs. The F-score of the NP chunker after all these changes is 94.51%, using the TnT tagger on the test corpus before chunking it.
Finally, yet another Transformation List was learnt, to bracket complex **Verb Phases**, such as sequences of auxiliaries and verbs, or verbs with adverbs inside (e.g *to accurately obtain*). The training corpus was also obtained from the tagged version of the Penn Treebank. After bracketing, the Verb Phrases are analysed and annotated with information about tense, person, number, voice (active or passive), and other information.
For Spanish, a Noun Phrase Chunker, trained on the UAM Treebank (Moreno et al., 2003), is also available.

3.5. Parsing

The parser (only for English) is based on several handcrafted rules that have been written for identifying subject-verb and verb-object relationships, as well as some prepositional phrase attachment in cases which are not ambiguous. This shallow parser is able to identify the subject-verb and the verb-object relations for around one half of the verbs in the texts analysed. Ambiguous pp-attachment is not currently handled.

3.6. Other modules

The suite was recently extended with a simple word-sense disambiguation algorithm based on the Lesk algorithm (Lesk, 1986), and a headline generation module (Alfonseca et al., 2004).

4. Future work

In the future, the suite is expected to be extended with the following capabilities:

- More efficient modules for Word Sense Disambiguation and Summarisation that, being the last additions to the suite, still have room for improvement.
- An anaphora resolution module that is already under construction.
- A more robust ontology-based knowledge-representation system. It will continue supporting WordNet as a dictionary of word senses for the Word Sense Disambiguation module, and for the generation of summaries, but it will be possible to use instead other domain-dependent or user-dependent ontologies.
- Tools for semi-automatic ontology acquisition from corpora, which are currently in the stage of development.
- Addition of all the modules that are currently missing for Spanish, and a possible extension to other languages.
- Finally, if possible, we would like to implement a cleaner, easy to compile version to be released as open-source.

5. References

E. Alfonseca and S. Manandhar. 2002. Extending a lexical ontology by a combination of distributional semantics signatures. In *Knowledge Engineering and Knowledge Management*, volume 2473 of *Lecture Notes in Artificial Intelligence*, pages 1–7. Springer Verlag.

E. Alfonseca and M. Ruiz-Casado. 2005. Learning sure-fire rules for named entities recognition. In *Proceedings of the International Workshop in Text Mining Research, Practice and Opportunities, in conjunction with RANLP conference*, Borovets, Bulgary.

E. Alfonseca, J. M. Guirao, and A. Moreno-Sandoval. 2004. Description of the UAM system for generating very short summaries at DUC-2004. In *DUC-2004*.

E. Alfonseca, A. Moreno-Sandoval, J. M. Guirao, and M. Ruiz-Casado. 2005. Wraetlic user guide version 2.0.

K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. 2004. Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering*, 10:349–373.

T. Brants. 2000. Tnt – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000*, Seattle, WA, U.S.A.

C. Brew, D. McKelvie, R. Tobin, H. Thompson, and A. Mikheev. 2000. The XML library LT XML version 1.2. user documentation and reference guide.

X. Carreras, I. Chao, L. Padró, and M. Padró. 2004. Freeling: An open-source suite of language analyzers. In *Proceedings of the LREC-2004*, Portugal.

H. Cunningham, K. Humphreys, R. Gaizauskas, and Y. Wilks. 1997. GATE–a TIPSTER-based general architecture for text engineering. In *Proceedings of the TIPSTER Text Program (Phase III) 6 Month Workshop*. DARPA, Morgan Kaufmann, California.

H. Cunningham, K. Bontcheva, V. Tablan, and Y. Wilks. 2000. Software infrastructure for language resources: a taxonomy of previous work and a requirements analysis. In *Proceedings of LREC-2002*, Athens.

W. Drozdzynski, H.-U. Krieger, J. Piskorski, and U. Schäfer. 2005. SProUT – a general-purpose NLP framework integrating finite-state and unification-based grammar formalisms. In *5th International Workshop on Finite-State Methods and Natural Language Processing*, Helsinki, Finland.

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. 1995. University of Sheffield: Description of the lasie system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220. Morgan Kauffmann.

R. Grishman. 1996. Tipster architecture design document version 2.2.

R. Grishman. 2005. Java extraction toolkit.

N. Ide and L. Romary. 2003. Outline of the international standard linguistic annotation framework. In *Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model Right*, pages 1–5, Sapporo.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries. In *Proceedings of the 5th International Conference on Systems Documentation*, pages 24–26.

E. Loper and S. Bird. 2005. NLTK: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 62–69, Somerset, NJ. Association for Computational Linguistics.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

D. McKelvie, C. Brew, and H. S. Thompson. 1997. Using SGML as a basis for data intensive natural language processing. *Computers and the Humanities*, 31(5):367–388.

A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazeteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway.

A. Mikheev. 2002. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):245–288.

A. Moreno, S. López, R. Grishman, and F. Sánchez, 2003. *Developing a syntactic annotation scheme and tools for a Spanish treebank. In Anne Abeille (Ed.) Treebanks: Building and Using Parsed Corpora*, pages 149–163. Kluwer Academic.

A. Moreno-Sandoval, G. de la Madrid, M. Alcántara, A. Gónzalez, J. M. Guirao, and R. de la Torre. 2005. The spanish corpus. In *C-ORAL-ROM: Integrated Reference Corpora for Spoken Romance Languages*, number 15 in Studies in Corpus Linguistics, pages 135–161. John Benjamins.

OpenNLP. http://opennlp.sourceforge.net/.

D. Pérez, A. Gliozzo, C. Strappavara, E. Alfonseca, P. Rodríguez, and B. Magnini. 2005. Automatic assessment of students' free-text answers underpinned by the combination of a BLEU-inspired algorithm and LSA. In *Proceedings of FLAIRS-2005*.

G. Petasis, V. Karkaletsis, G. Paliouras, I. Androutsopoulos, and C. Spyropoulos. 2002. Ellogon: A new text engineering platform. In *Proceedings of LREC 2002*, pages 72–78, Las Palmas, Spain.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Third ACL Workshop on Very Large Corpora*, pages 82–94. Kluwer.

SAIC. 1998. *Proceedings of the Seventh Message Understanding Conference, MUC-7*. http://www.muc.saic.com.

S. Sekine. 2002. Oak system version 0.1.

N. K. Simkins. 1994. An open architecture for language engineering. In *First Language Engineering Convention*, Paris, France.

H. S. Thompson and D. McKelvie. 1997. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML Europe '97*, Barcelona, Spain.

F. Wolinski, F. Vichot, and O. Gremont. 1998. Producing NLP-based on-line contentware. In *Natural Language and Industrial Applications*, Moncton, Canada.