An open-source shallow-transfer machine translation toolbox: consequences of its release and availability

Carme Armentano-Oller, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Boyan Bonev, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez

Transducens group, Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant

carmentano@dlsi.ua,es, acorbi@dlsi.ua.es, mlf@ua.es, mginesti@dlsi.ua.es, bib@alu.ua.es, sortiz@dlsi.ua.es, japerez@dlsi.ua.es, gema@internostrum.com, fsanchez@dlsi.ua.es

Abstract. By the time Machine Translation Summit X is held in September 2005, our group will have released an open-source machine translation toolbox as part of a large government-funded project involving four universities and three linguistic technology companies from Spain. The machine translation toolbox, which will most likely be released under a GPL-like license includes (a) the open-source engine itself, a modular shallow-transfer machine translation engine suitable for related languages and largely based upon that of systems we have already developed, such as interNOSTRUM for Spanish—Catalan and Traductor Universia for Spanish—Portuguese, (b) extensive documentation (including document type declarations) specifying the XML format of all linguistic (dictionaries, rules) and document format management files, (c) compilers converting these data into the high-speed (tens of thousands of words a second) format used by the engine, and (d) pilot linguistic data for Spanish—Catalan and Spanish—Galician and format management specifications for the HTML, RTF and plain text formats. After describing very briefly this toolbox, this paper aims at exploring possible consequences of the availability of this architecture, including the community-driven development of machine translation systems for languages lacking this kind of linguistic technology.

1. Introduction

By the time Machine Translation Summit X is held in September 2005, our group will have released an open-source machine translation toolbox as part of a large government-funded project involving four universities and three linguistic technology companies from Spain.[1] The machine translation toolbox, which will most likely be released under a GPL-like license includes:

(a) the open-source engine itself, a modular shallow-transfer machine translation engine suitable for related languages and largely based upon that of systems we have already developed, such as interNOSTRUM (Canals-Marote et al. 2001) for Spanish—Catalan and Traductor Universia (Garrido-Alenda et al. 2004) for Spanish—Portuguese,

(b) extensive documentation (including document type declarations) specifying the XML format of all linguistic (dictionaries, rules) and document format management files,

(c) compilers converting these data into the high-speed (tens of thousands of words a second) format used by the engine, and

(d) pilot linguistic data for Spanish—Catalan and Spanish—Galician and format management specifications for the HTML, RTF and plain text formats.

1 Eleka Ingeinaritza Linguistikoa (coordinator), Elhuyar Fundazioa, Imaxin Software, Euskal Herriko Unibertsitatea, Universidade de Vigo, Universitat Politècnica de Catalunya and Universitat d'Alacant.

After describing briefly this toolbox[2], this paper aims at exploring possible consequences of the availability of this architecture (or similar ones), including the community-driven development of machine translation systems for languages lacking this kind of linguistic technology.

Almost all existing machine translation (MT) programs are mostly commercial or use proprietary technologies, which makes them very hard to adapt to new usages, and use different technologies across language pairs, which makes it very difficult to integrate them in a single multilingual content management system. As Beninatto (2003) puts it, "Commercial, government and academic groups spread throughout the world spend a lot of effort and money into developing the perfect [MT] system. Their efforts are seldom leveraged against or in support of each other." After acknowledging that "open source development has given life to excellent systems and applications", Beninatto (2003) concludes that "it is high time for the language technology industry to join forces and establish guidelines for the development of the 'Linux of machine translation'".

Indeed, one of the main novelties of the toolbox described here is that it will be released under an open-source license[3] (together with pilot linguistic data for Spanish—Catalan and Spanish—Galician) and will be distributed free of charge. This means that anyone having the necessary computational and linguistic skills will be able to adapt or enhance it to produce a new MT system, even for other pairs of related languages. The toolbox will likely be available by the time MT Summit X is held.

2 A more detailed description may be found in Corbí-Bellot et al. (2005).

3 The license has still to be determined. Most likely, the toolbox will be released under the GPL license. This is not the first open-source machine translation project: There are other projects such as Traduki (http://traduki.sourceforge.net), GPLTrans (http://www.translator.cx), or Linguaphile (http://linguaphile.sourceforge.net/); however, this will be the first project to release a real, general purpose, system which is based upon the experience of systems already being used on a daily basis by thousands of users.

The MT toolbox concerned here uses finite-state transducers for lexical processing, hidden Markov models for part-of-speech tagging, and finite-state based chunking for structural transfer, and is largely based upon that of systems already developed by the Transducens group such as interNOSTRUM[4] (Spanish—Catalan, Canals-Marote et al. 2001) and Traductor Universia[5] (Spanish—Portuguese, Garrido-Alenda et al. 2004); these systems are publicly accessible through the net and used on a daily basis by thousands of users.

We expect that the introduction of a unified open-source MT architecture will ease some of the mentioned problems (having different technologies for different pairs, closed-source architectures being hard to adapt to new uses, etc.); it will also foster the development of MT systems for new language pairs not addressed by major companies or academic institutions. Finally, it will also help shift the current business model from a licence-centred one to a services-centred one, and favour the interchange of existing linguistic data through the use of the XML-based formats defined in this project.

The paper is organized as follows: section 2 gives a brief description of the toolbox: the engine (sec. 2.1), the formats defined for the encoding of linguistic data (sec. 2.2), and the compilers used to convert these data into an executable form (sec. 2.3); section 3 describes types of language pairs which may be benefited by the release of this toolbox; section 4 gives hints as to how the toolbox may be used by communities of developers to build machine translation systems for new language pairs; finally, we give some concluding remarks (sec. 5).

2. The MT toolbox

2.1. The engine

The MT strategy used in the system has already been described in detail (Canals-Marote et al. 2001; Garrido-Alenda et al. 2004); a sketch will be given here. The engine is a classical shallow-transfer or transformer system consisting of an 8-module assembly line; we have found that the shallow-transfer strategy is sufficient to achieve a reasonable translation

4 http://www.internostrum.com/

5 http://traductor.universia.net/

quality between related languages: while, for these languages, a rudimentary word-for-word

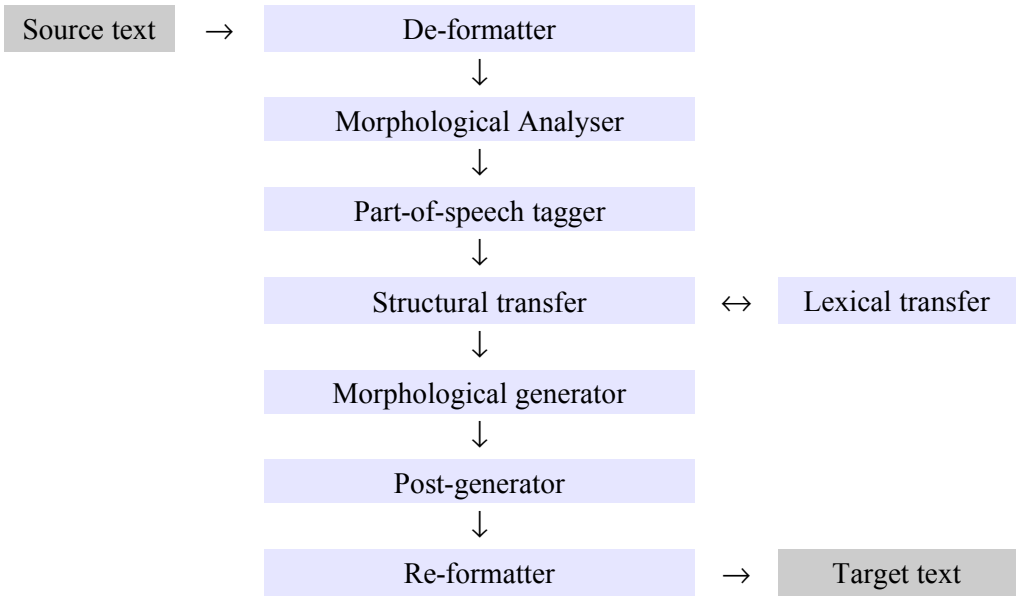

Figure 1: The eight modules of the open-source shallow-transfer machine translation engine.

MT model may give an adequate translation for, say, 50%—75% of the text, the addition of homograph disambiguation, management of contiguous multi-word units, and local reordering and agreement rules may raise the fraction of adequately translated text above 90%. This is the approach used in the engine presented here.

To ease diagnosis and independent testing, modules communicate between them using text streams. This allows for some of the modules to be used in isolation, independently from the rest of the MT system, for other natural-language processing tasks. The modules are shown in figure 1.

Most of the modules are capable of processing tens of thousands of words per second on current desktop workstations; only the structural transfer module lags behind at several thousands of words per second.

Here is a brief description of the modules:

- The **de-formatter** separates the text to be translated from the format information (RTF, HTML, etc.). Format information is encapsulated so that the rest of the modules treat it as blanks between words. The de-formatter is automatically generated by a compiler (sec. 2.3) from a file specifying the formatting rules for each type of document.
- The **morphological analyser** tokenizes the text in *surface forms* (lexical units as they appear in texts) and delivers, for each surface form, one or more *lexical forms* consisting of *lemma*, *lexical category* and morphological inflection information. The system is capable of dealing with contractions and fixed-length multi-word lexical units (either invariable or inflected). The module reads in a binary file compiled by the lexical transformation compiler (section 2.3) from a source-language morphological dictionary (see section 2.2).
- **Part-of-speech tagger:** a sizeable fraction of surface forms (for instance, about 30% in Romance languages) are homographs, that is, ambiguous forms for which the morphological analyser delivers more than one lexical form. The part-of-speech tagger chooses one of them, according to the lexical forms of neighbouring words. When translating between related languages, ambiguous surface forms are one of the main sources of errors when incorrectly solved. The part-of-speech tagger reads in a file containing a hidden Markov model (HMM) which has been trained on representative source-language texts (using an open-source training program in the toolbox, see section 2.3). The behaviour of both the part-of-speech tagger and the training program are both controlled by a tagger definition file (see section 2.2).

- The **lexical transfer module** is called by the structural transfer module (see next section); it reads each source-language lexical form and delivers a corresponding target-language lexical form. Before execution, the module reads in a binary file compiled from a bilingual dictionary (see section 2.2). The dictionary contains a single equivalent for each source-language entry; that is, no word-sense disambiguation is performed. For some words, however, multi-word entries are used to safely select the correct equivalent in frequently-occurring fixed context, an approach used with very good results in Traductor Universia and interNOSTRUM. (Canals-Marote et al. 2001; Garrido-Alenda et al. 2004).
- The **structural transfer module** uses finite-state pattern matching to detect (in the usual left-to-right, longest-match way) fixed-length patterns of lexical forms (*chunks* or *phrases*) needing special processing due to grammatical divergences between the two languages (gender and number changes to ensure agreement in the target language, word reorderings, lexical changes such as changes in prepositions, etc.) and performs the corresponding transformations. This module is compiled —using a structural transfer rule compiler (see section 2.3)— from a transfer rule file (see section 2.2).
- The **morphological generator** delivers a target-language surface form for each target-language lexical form, by suitably inflecting it. The module reads in a binary file compiled —using a generic lexical transformation compiler (see section 2.3)— from a target-language morphological dictionary (see section 2.2).
- The **post-generator** performs orthographical operations such as contractions and insertion of apostrophes. The module reads in a binary file compiled by the generic lexical transformation compiler (see section 2.3) from a rule file expressed as a string transfromation dictionary (section 2.2). The post-generator is usually *dormant* (just copies the input to the output) until a special alarm symbol contained in some target-language surface forms *wakes* it *up* to perform a particular string transformation if necessary; then it goes back to *sleep*.
- Finally, the **re-formatter** restores the format information encapsulated by the de-formatter into the translated text and removes the encapsulation sequences used to protect certain characters in the source text. The de-formatter is also automatically generated by a compiler (the format management compiler, see section 2.3) from a file specifying the formatting rules for each type of document.

2.2. Formats for linguistic data

An adequate documentation of the code and auxiliary files is crucial for the success of open-source software. In the case of a MT system, this implies carefully defining a systematic format for each source of linguistic data used by the system. The formats used by this architecture (which will not be described in detail for lack of space) are based on those used by interNOSTRUM and Traductor Universia. We have chosen to base them upon a widespread platform, XML (World Wide Web Consortium, 2004) for interoperability; in particular, for easier parsing, transformation, and maintenance of linguistic data. The XML formats for each type of linguistic data are defined through conveniently-designed XML document-type definitions (DTDs) and will also be widely documented. The use of XML allows for a *declarative* approach to machine translation development, which makes it easy for linguistic developers to focus on the linguistic nature of translation between the two languages involved.

There are five main formats for data:

1. **Dictionaries**. A unified format is used for the monolingual *morphological dictionaries* used to specify the source-language morphological analyser and the target language morphological generator, the bilingual dictionaries used to specify the lexical transfer module, and the rules describing the orthographical transformations performed by the post-generator. In all of them, linguistic regularities may be easily encoded to avoid long enumerations of forms; for example, morphological dictionaries may use inflection paradigms to encode regularities in inflection.
2. The **tagger definition file** is used to define the behaviour of the part-of-speech tagger

both when it is being trained on a source-language corpus and when it is running as part of the MT system. It specifies (a) the categories that it will distinguish, which are defined by grouping the finer part-of-speech tags delivered by the morphological analyser; (b) linguistic constraints to forbid or enforce certain sequences of part-of-speech tags, and (c) priority lists to decide which fine part-of-speech tag to pass on to the structural transfer module when the coarse part-of-speech tag delivered by the tagger contains more than a fine tag.

3. The **training corpora** may be of two types: ambiguously tagged corpora, in which each surface forms appears with all the possible part-of-speech tags delivered by the morphological analyser, and unambiguously tagged corpora, where the correct tag has been manually selected by an expert. Both kinds of corpora use XML tags as defined by a suitable DTD. The first kind of corpora is used for unsupervised training (using an expectation-maximization algorithm) and the second one is used for supervised training (estimating probabilities from frequencies in a maximum-likelihood scheme).

4. The **structural transfer rule files**: they contain pattern—action rules describing what has to be done for each pattern. Using a declarative notation such as XML for the action (procedural) part means stretching it a bit; we have, however, found a reasonable design for it, based on the language used in the corresponding module of interNOSTRUM and Traductor Universia, which was defined in detail by Garrido-Alenda and Forcada (2001).

5. The **format management files.** A format management file describes the behaviour of the de-formatter and the re-formatter for a certain file format by defining how to encapsulate format information and isolate it from the text to be translated (de-formatter) and how to integrate it back (re-formatter). While this cannot properly be considered a set of linguistic data, it is grouped here with the rest of the linguistic data specifications in the system. We will provide pilot files for HTML, RTF and plain text, but files for additional formats (or even programming languages in the case of software localization) may be easily implemented.

2.3. Compilers

The toolbox will provide compilers to convert the linguistic data into the corresponding efficient form used by each of the modules of the engine. Four compilers are used in this project:

- The **lexical processor compiler**: The four lexical processing modules (morphological analyser, lexical transfer, morphological generator, post-generator) are currently being implemented as a single program which reads binary files containing a compact and efficient representation of a class of finite-state transducers (*letter transducers*, Roche & Schabes 1997); in particular, *augmented* letter transducers (Garrido-Alenda et al. 2002). These binaries are generated in seconds from XML dictionaries (specified in section 2.2) using a single compiler written in C++. Fast compilation makes linguistic data development much easier, because the effect on the whole system of changing a rule or a lexical item may be tested almost immediately.
- The **structural transfer compiler** is simply an XSLT stylesheet which, executed on a standard XSLT processor, reads in the XML file with structural transfer rules and produces a `lex` module which is then compiled into C, and eventually into an executable module.
- The **format management file compiler** is also an XSLT stylesheet transforming the XML specification of the format management modules (de-formatter and re-formatter) and generating a `lex` file which is eventually turned into an executable module.
- The **tagger training programs**, written in C++, are not compilers but may be considered as such in this project because they read in linguistic data (a tagger definition file, a training corpus, and, optionally, a source-language morphological dictionary) and output a part-of-speech tagger module. After training two output files are generated: one with the ambiguity classes found in the morphological dictionary, and another one with the transition and emission probabilities of the hidden Markov model. These are the files

read by the part-of-speech tagger during translation.

3. Benefited language pairs

As has been said above, the shallow-transfer MT toolbox described here is most suitable for morphosyntactically related language pairs having a small degree of divergence, usually because of having a common origin or because of belonging to the same language group. This still leaves a lot of interesting language pairs in the world. Among these, some (usually *national*[6] language pairs) have commercial systems available (such as Spanish—Catalan, Spanish—French or Spanish—Italian, just to name some). Leaving aside that the availability of an open-source MT toolbox could still spark the development of alternative MT systems for these major language pairs, there are still a variety of situations involving pairs of related languages which may be affected by the release of the toolbox. Here is a set of situations or *stories* which may serve as an illustration, without aiming at being exhaustive:

- Catalan (a medium-sized Romance language having about 6 million speakers) is spoken mainly in Spain, where has been recognized as co-official in some regions, but is also spoken in South-Eastern France and in the Sardinian city of l'Alguer (Alghero), Italy, where it is basically non-official, but there exist groups that struggle for its normality, especially groups asking for Catalan schooling of children. The opportunity to develop a freely-available Catalan—French or Catalan—Italian MT system could help Catalan improve its status in France and Italy, but would also connect it to two main *national* languages, allowing it to interact with them directly instead of through Spanish. The local governments of the Catalan-speaking areas could also promote the use the toolbox as an opportunity to build systems connecting Catalan to other major languages of the world, such as Portuguese (Portuguese being the *national* language of countries totalling hundreds of millions of people) or Romanian.
- Occitan (also called Gascon, Provençal, Aranese, Piamontese, etc.) is spoken mainly in France but also in parts of Italy and Spain. This language, one of the main literary languages in Medieval Europe, is reported to still have about a million speakers, but has almost no legal existence in France and Italy and a limited status of co-officiality in a very small part of Catalonia in Spain. There are groups, mainly in France, who want to increase the legal recognition of Occitan, with people prepared to build linguistic data for a French—Occitan, Occitan—French system which would be of great help to generate Occitan texts and to make Occitan texts understandable to speakers of French.
- The Italian—Sardinian or French—Corsican cases are also worth mentioning; a MT system for this pairs could be very beneficial to these smaller romance languages which have a very limited legal recognition in their countries.
- There exist pairs of related *national* languages which have no machine translation technology available to them, such as pairs of Slavic languages (Czech—Slovak, Serbo-Croatian—Slovenian, etc.), Scandinavian languages (Swedish—Danish, Norwegian Bokmål—Norwegian Nynorsk, etc.), or Bantu languages (Swahili [Tanzania]—Kirwanda [Rwanda], etc.).
- In addition to Spanish and the three main co-official languages (Basque, Galician and Catalan), there are three small Romance languages in Spain that would benefit from the availability of machine translation for them: Asturian, Aranese (a variety of Occitan) and Aragonese.

4. Community development of new MT systems

One of the possible ways in which the machine translation toolbox presented here may be used to generate a machine translation system for a new language pair is through communities of volunteers. As has been mentioned in the previous section, many languages far from normality or officialness have activist groups, usually in the education arena, which include people whose linguistic and translation skills would allow them to

6 We will use the (oversimplified) name *national* to refer to those languages which are official in the entire territory of an independent country, having into account that there are some countries which may be seen as formed by more than one nation (such as Spain), some of them having their own language.

collaborate in the creation of linguistic data (dictionaries and rules).

But language and translation skills and volunteered time, even if completely crucial in the case of languages lacking official support, are not enough: volunteer work should be coordinated by a smaller group of people who master the details of the MT toolbox presented here. Here are some ingredients of a possible way to organize such a project:

- Each language pair would have a coordinating team, that is, a small group of people mastering the toolbox, which would lead the project (see below). This coordinating team could optionally have a code captain (dealing with installation, maintenance and possible modifications of the code of the engine or the compilers) and a linguistic captain (responsible for the maintenance of linguistic data).
- A project server and website, which would serve both as the interface through which (registered) volunteers would contribute new linguistic data —for example, monolingual and bilingual dictionary entries through a form interface designed to elicit the necessary linguistic knowledge and generate XML dictionary data from it— and as a way for users in the linguistic community involved to download or execute the latest build (version) of the translator. The website would be administered by the coordinating team; ideally, the website should reside in a computer over which the coordinating team have complete control (installing software, adding users, etc.).
- A group of volunteers, ideally certified in some sense by the coordinating team to have the necessary linguistic and translation skills to make useful contributions to dictionaries.

A formula which can be worth exploring to *start* such a project may be some kind of *marathon* or *volunteer party* in which a group of volunteers physically get together (for example, during a weekend) to build linguistic data (for example, generating entries for the first few thousand most frequent words in a corpus, or keying in the entries in a bilingual pocket dictionary which is torn in similarly-sized portions which are given to each participant). The coordinating team would have to prepare a big room with enough computers, install the necessary software for the effort, and arrange for meals and basic lodging. This scheme was used recently, for instance, to localize OpenOffice.org 2.0 into Catalan.[7]

Admittedly, there are parts of the linguistic data that are more suitable for volunteer development than other. With a well-designed form interface capable of eliciting the linguistic knowledge of volunteers, it is possible to maintain simultaneously the dictionaries of the system (source-language morphological dictionary, target-language morphological dictionary and the bilingual dictionary). However, one can argue that the design of transfer rules or tagger definition files does not lend itself so easily to volunteer work (elicitation of user knowledge in these cases is a research topic on itself; see, for instance Sherematyeva and Nirenburg 2000, Font-Llitjós et al 2005).

5. Concluding remarks

This paper describes an open-source toolbox (will be released just about by the time MT Summit X is held) that may be used to generate shallow-transfer machine translation systems (for related languages) simply by coding linguistic and format management data in XML-based stanadard formats. The paper goes on to explore the consequences it may have in a variety of language-pair settings, especially those involving languages which are not the official languages of large countries.

Acknowledgements: The development of the toolbox is funded by project FIT-340101-2004-3 (Spanish Ministry of Industry, Commerce and Tourism).

6. References

BENINATTO, ROBERTO. (2003) “The Automated Translation Manifesto: A Vision for Automated Translation and its Role in Communications”, Proc. of the 23th Internationalization and Unicode Conference (Prague, 24—26 March), available from http://www.commonsenseadvisory.com.

CANALS-MAROTE, R., A. ESTEVE-GUILLÉN, A. GARRIDO-ALENDA, M.I. GUARDIOLA-SAVALL, A. ITURRASPE-BELLVER, S. MONTSERRAT-BUENDIA, S. ORTIZ-ROJAS, H. PASTOR-PINA, P.M. PÉREZ-ANTÓN, M.L. FORCADA (2001). "The Spanish-Catalan machine translation system interNOSTRUM", in B. Maegaard, ed., *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, 73-76.

7 See http://ca.OpenOffice.org

Corbí-Bellot, A.M., Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, Kepa Sarasola (2005) “An Open-Source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain”, Proceedings of EAMT 2005 (Budapest, 30-31 May 2005).

Font-Llitjós, A., Jaime G. Carbonell, Alon Lavie (2005) “A Framework for Interactive and Automatic Refinement of Transfer-based Machine Translation”, Proceedings of EAMT 2005 (Budapest, 30-31 May 2005).

Garrido, A., Amaia Iturraspe, Sandra Montserrat, Hermínia Pastor, Mikel L. Forcada (1999). "A compiler for morphological analysers and generators based on finite-state transducers", *Procesamiento del Lenguaje Natural*, **25**, 93-98.

Garrido-Alenda, A., M.L. Forcada (2001). "MorphTrans: un lenguaje y un compilador para especificar y generar módulos de transferencia morfológica para sistemas de traducción automática", *Procesamiento del Lenguaje Natural*, **27**, 157-162.

Garrido-Alenda, A. Mikel L. Forcada, Rafael C. Carrasco (2002). "Incremental construction and maintenance of morphological analysers based on augmented letter transducers", in Mitamura, T., Nyberg, E., ed., *Proceedings of TMI 2002 (Theoretical and Methodological Issues in Machine Translation, Keihanna/Kyoto, Japan, March 2002)*, 53-62.

Garrido-Alenda, A., Patrícia Gilabert Zarco, Juan Antonio Pérez Ortiz, Antonio Pertusa-Ibáñez, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Míriam A. Scalco, Mikel L. Forcada (2004). "Shallow parsing for Portuguese-Spanish Machine Translation", in Branco, A. and Mendes, A., Ribeiro, R., *Language technology for Portuguese: shallow processing tools and resources* , 135-144.

Roche, E., Schabes, Y. (1997). "Introduction", in Roche, E., Schabes, Y., *Finite-state language processing* , 1-65.

Sherematyeva, Svetlana; S. Nirenburg. (2000). ”Towards a Universal Tool For NLP Resource Acquisition”. Proceedings of The Second International Conference on Language Resources and Evaluation (Greece, Athens, May 31 -June 3, 2000).

World Wide Web Consortium (2004). "Extensible Markup Language (XML)", `http://www.w3.org/XML/`.