The Spanish⇔Catalan machine translation system *interNOSTRUM*

R. Canals−Marote, 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, and M.L. Forcada,

interNOSTRUM
Departament de Llenguatges i Sistemes Informàtics,
Universitat d'Alacant, E−03071 Alacant
E−mail: `mlf@dlsi.ua.es`

Abstract

This paper describes *interNOSTRUM*, a Spanish⇔Catalan machine translation system currently under development that achieves great speed through the use of finite−state technologies (so that it may be integrated with internet browsing) and a reasonable accuracy using an advanced morphological transfer strategy (to produce fast translation drafts ready for light postedition).

Keywords: Spanish, Catalan, finite−state, internet browsing, morphological transfer.

1 Introduction

This paper describes a Spanish⇔Catalan machine translation system, *interNOSTRUM*. The main reason for the demand of translations from Spanish (official language of Spain) into Catalan is the impulse toward 'linguistic normalization' in the Catalan−speaking areas (10 million inhabitants, about 6 million speakers) where Catalan was receding and where the language is now co−official; translation from Catalan into Spanish (mainly for assimilation purposes) allows the access to documents in Catalan by Spanish−speaking people in Spain or Latin America. The *interNOSTRUM* system is still under development and a prototype has been serving the Universitat d'Alacant, a medium−sized university, and the Caja de Ahorros del Mediterráneo (CAM), one of the largest savings banks in Spain, for about two years now, and a test version is freely available at http://internostrum.com. These two institutions started and currently fund this three−year project (1999−2001), which has a staff of two linguists and four computer engineers. Even though translation accuracy and vocabulary coverage can still be much improved (especially in the Catalan to Spanish version), the speed of the system −−thousands of words per second or tens of millions of words per day on a 1999−model desktop machine acting as an Internet server−− has prompted its use as a system to obtain instantaneous rough translations that are relatively easy to turn into publishable documents and as an aid to information assimilation during internet browsing (accesses to our server have been steadily increasing every month; we had 120,000 visits in March 2001). These speeds are achieved through the use of finite−state technology (Roche and Schabes, 1997) in most of its modules.

2 Current prototype and future versions

As has been said, *interNOSTRUM* is not yet a finished product; however, it has been available for some time now. This is because two of the basic objectives of our project have been, first, to generate an operational version of *interNOSTRUM* as soon as possible (Spanish−to−Catalan system launched November 1999) and, second, to make the latest stable version available as soon as it is ready. These are the main reasons for its current configuration as a single Internet server instead of as distributable software.

Currently, *interNOSTRUM* translates ANSI, RTF (Microsoft's Rich Text Format) and HTML (HyperText Markup Language) texts from Castillian Spanish to the central or Barcelona variety of Catalan and vice−versa. A version generating and accepting the València and the Balearic varieties should be ready in September 2001. The Catalan to Spanish version is very recent and therefore it is less satisfactory as to vocabulary coverage and accuracy. Error rates (measured as the number of words that have to be inserted, deleted or substituted per 100 words to render the target text acceptable) range around 5% in the Spanish−Catalan direction when newspaper text is translated and are somewhat worse in the Catalan−Spanish direction.

2.1 Platform

interNOSTRUM currently runs on Linux using Apache and PHP3 and may be accessed through a public Internet server (http://internostrum.com); an internal Windows 2000−Internet Information Services version of the server is being used inside the CAM. *interNOSTRUM* consists of eight modules that run in parallel and communicate through text channels (the use of human−readable text channels (pipelines) allows for an easy diagnosis of many problems and is a very efficient alternative in Linux (Unix) implementations). Six of the eight modules are automatically generated from the corresponding linguistic data using compilers written with the aid of yacc and lex, which are standard in Unix environments (see table 1); this feature makes *interNOSTRUM* easily extensible to other languages. The current speed of the system is in the order of 1,000 wps (words per second) on a standard 1999 desktop PC (a 400−MHz Pentium II PC).

2.2 Machine translation strategy

interNOSTRUM is a classical indirect machine translation system using an advanced morphological transfer strategy (similar to a *transformer architecture*, Arnold (1993) or *direct system*, Hutchins and Somers (1992) analogous to the one used in commercial PC−based machine translation systems, such as Transparent Technologies' Transcend RT, early versions of Globalink's Power Translator, and Softissimo's Reverso (Mira i Gimènez and Forcada, 1998; Forcada 2000). As said above, *interNOSTRUM* has eight modules (see figure 1): a deformatting module (which separates text from formatting information), two analysis modules (morphological analyser and part−of−speech tagger), two transfer modules (bilingual dictionary module and pattern processing module), two generation modules (morphological generator and postgenerator) and the reformatting module (which integrates the original formatting information with the text).

2.2.1 Modules based on finite−state technology

Four of the modules in *interNOSTRUM*, namely, the *morphological analyser*, the *bilingual dictionary module*, the *morphological generator*, and the *postgenerator* are based on *finite−state transducers* (FSTs) (Roche and Schabes 1997). This allows for processing speeds on the order of 10,000 wps, which are practically independent of the size of the dictionaries. FSTs read their input symbol by symbol; each time a symbol is read, they move to a new state, and write one or more output symbols.

The morphological analyser, which is automatically generated (Garrido et al., 1999) from a *morphological dictionary* (MD) for the source language (SL). The MD contains the lemmas (canonical or base forms for inflected words), the inflection paradigms, and their mutual relationships. The subprogram reads the text or *surface* forms (SF) and writes, for each surface form, one or more *lexical forms* (LF) consisting of a lemma, a part of speech, and inflection information.

The bilingual dictionary module is called by the pattern processing module (see below); it is automatically generated from a file that contains the bilingual correspondences. The program reads a SL LF and writes the corresponding target−language (TL) LF.

The morphological generator performs basically the reverse of morphological analysis, but applied to the TL. The morphological generator is generated from a MD for the TL.

The postgenerator: Those SF involved in apostrophation and hyphenation (such as clitic pronouns, articles, some prepositions, etc.) activate this module which is otherwise asleep. The postgenerator is generated from a file containing the corresponding rules for the TL.

The division of a text in words has some nontrivial aspects. On the one hand, there are a number of word groups that cannot be translated word for word and may be treated as fixed−length *multiword units* (MWU); they are continuously being incorporated to the morphological and bilingual dictionaries. The corresponding modules have support for fixed and variable (inflected) multiword units, which are used to avoid, where possible, translation problems due to homography, polysemy or non−compositional structures such as idioms or collocations. Examples: Sp. *con cargo a* → Cat. *a càrrec de* (''at the expense of''); Sp. *por adelantado* → Cat. *per endavant* (''in advance''); Sp. ***echar*** *de menos* → Cat. ***trobar*** *a faltar* (''to miss [someone]''; Sp. ***tener*** *que* + infinitive → Cat. ***haver*** *de* + infinitive (''have to''). In the last two examples (a modal construction and an idiom), the MWU has a variable element that may be inflected (in boldface). On the other hand, combinations of certain verb forms and enclitic pronouns are written in Spanish as a single word; these combinations occur with ortographical transformations such as accent marks or loss of consonants: Sp. *dámelo* = *da+me+lo* → Cat. *dóna+me+lo* = *dóna−me'l* (''give it to me!''); Sp. *presentémonos* = *presentemos+nos* → Cat. *presentem+nos* = *presentem−nos* (''let us introduce ourselves''); these are dealt with by the morphological analyser.

2.2.2 The part−of−speech tagger

Most lexical ambiguities fall into two main groups: *homography* (when a SF has more than one LF or analysis) and *polysemy* (when the SF has a single LF but the lemma may have more than one interpretation). The lexical disambiguation module or *part−of−speech tagger* uses a hidden Markov model based on bigrams and trigrams (sequences of two and three lexical categories) to solve homographs featuring a categorial ambiguity. The model's parameters reflect the statistics of word distribution observed in a reference text corpus; the tagger computes on the fly the most likely disambiguation of a sentence. We are currently fine−tuning the tagset used and building a larger training corpus to improve the performance of this module. The current tagset has about 60 tags for each language , and is different from general−purpose tagsets in that it establishes specialized distinctions which are relevant for translation. For example, the Spanish tagset distinguishes subjunctive and indicative forms of verbs in order to disambiguate forms such as *salen* which may be either the 3rd person plural of the present indicative of *salir* ("to go out") or the 3rd person plural of the present subjunctive of *salar* ("to salt"). The few errors occurring in certain *difficult* but *frequent* Spanish homographs, such as *una* (article/verb), *para* (verb/preposition), and *como* (conjunction/verb) constitute one of the main contributors to the current error rate in *interNOSTRUM*. Fortunately, other frequent homographs are easier to disambiguate. Similar problems are found when analysing Catalan. Polysemy is not treated (only in some cases through the use of fixed−length multiword units representing collocations): the bilingual dictionary gives always the same human−chosen equivalent for each lemma; we have found that errors due to polysemy are much less frequent than those due to homography. Polysemic words will be dealt with through the use of a *controlled Spanish* biased toward banking and administration applications (see section 3).

2.2.3 The pattern processing module

In spite of the great similarity between Spanish and Catalan, there are still a number of important grammatical divergences: gender and number divergences which affect agreement −−Sp. *la deuda contraída* (fem.) → Cat. *el deute contret* (masc.; ''the assumed debt'')−−; relative constructions using *cuyo* (''whose''), absent in Catalan −−Sp. *la cuenta cuyo titular es el asegurado* → Cat. *el compte el titular del qual és l'assegurat* (Engl. ''the account whose owner is the insured person''), or prepositional changes −−Sp. *en Londres* → Cat. *a Londres* (Eng. ''in London''). These divergences have to be treated using suitable grammatical rules. *interNOSTRUM* uses a solution which may also be found in commercial MT systems (Mira i Gimènez and Forcada, 1998; Forcada 2000). It is based on the detection and treatment of predefined sequences of lexical categories (*patterns* or *chunks*) which may be seen as rudimentary phrase−structure constructs: for example, **art.−noun** or **art.−noun−adj.** are two possible valid noun phrases. Those sequences known to the program constitute its *pattern catalog*. This module works as follows:

- The text (morphologically analysed and disambiguated) is read left to right, one LF at a time.
- The module searches, starting at the current position in the sentence, for the longest LF sequence that matches a pattern in its pattern catalog (for example, if the text starting in the current position is ''una señal inequívoca...'' (''an unmistakable signal''), it will choose **art.−noun−adj.** instead of **art.−noun**).
- The module operates on this pattern (to propagate gender and number agreement, to reorder it, to make lexical changes) following the rules associated to the pattern.
- Then, the pattern processing module continues immediately after the pattern just processed (it does not visit again any of the LFs on which it has operated).

When no pattern is detected in the current position, the program translates one LF word for word and restarts at the following LF. ''Long−range'' phenomena such as subject−verb agreement require the propagation of information from one pattern to the following ones, which is supported as interpattern state information.

The pattern processing module is automatically generated from a source file containing rules that specify the patterns and the associated actions. This is the slowest module (still above 1,000 wps), compared to the 10,000 wps of the rest of the modules. The current catalog only contains about two dozen patterns.

2.3 The deformatter and the reformatter

Both modules are written in `lex` and `C++`, the reformatter being much simpler. There are three versions of each module: the ANSI version, the RTF version and the HTML version. The deformatter is used to identify and separate formatting commands from text to be translated. Formatting information, inline images, etc. are encapsulated (in double square brackets ''`[[...]]`'') to form *superblanks*, so that the remaining modules see them as whitespace between words (large segments of formatting material (>8 kB) are written to temporary files whose unique name is sent to the remaining modules instead of the data itself); when the linguistic modules produce translations having more or less words than the original, a queue manager ensures that superblanks are not lost. A special version of the deformatter converts URLs in ''`<A HREF`'' tags so that they are routed through *interNOSTRUM* to allow for real−time translation during browsing. We also offer support for many kinds of frame−based pages.

3 Projected support tools for *interNOSTRUM*

We are currently working on three support tools: (a) a *style assistant* to help authors of Spanish texts avoid many difficult ambiguities using the syntactical, lexical and style rules specified in a controlled Spanish; (b) a *preedition assistant*, for the manual disambiguation of problematic words and structures, by clicking on them to get a menu of options (helpful when the statistical strategy used by the program is unable to make the right choice); and (c) a *postedition assistant*, in which the author will be able to click on a target−language word when he or she suspects that it is an incorrect translation and will allow him or her to substitute it by an alternative among those compatible with the original text.

4 Concluding remarks

We have presented *interNOSTRUM*, a Spanish−Catalan machine translation system currently under development that achieves great speed through the use of finite−state technologies and a reasonable accuracy using an advanced morphological transfer strategy. The system is available as an internet server and it is being used mainly to obtain draft translation of Spanish documents into Catalan and to browse through Catalan internet servers in Spanish.

Acknowledgements: Work supported by the Caja de Ahorros del Mediterráneo, the Vice−Rectorate for Information Technologies of the Universitat d'Alacant and more recently by the Spanish Comisión Interministerial de Ciencia y Tecnología through project TIC2000−1599−C02−02.

Bibliography

Arnold, D. (1993). Sur la conception du transfert. In Bouillon, P. and Clas, A., editors, *La traductique*, pages 64−76. Presses Univ. Montréal, Montréal.

Forcada, M.L. (2000). Learning machine translation strategies using commercial systems: discovering word−reordering rules. In *MT2000: Machine Translation and Multilingual Applications in the New Millenium (Exeter, UK, November 18−20, 2000)*, pages 7.1−7.8.

Garrido, A., Iturraspe, A., Montserrat, S., Pastor, H., and Forcada, M. (1999).A compiler for morphological analysers and generators based on finite−state transducers. *Procesamiento del Lenguaje Natural*, (25):93−98.

Hutchins, W. and Somers, H. (1992). *An Introduction to Machine Translation*. Academic Press.

Mira i Gimènez, M. and Forcada, M. L. (1998). Understanding PC−based machine translation systems for evaluation, teaching and reverse engineering: the treatment of noun phrases in Power Translator. *Machine Translation Review (British Computer Society)*, 7:20−27. (available at http://www.dlsi.ua.es/~mlf/mtr98.ps.Z).

Roche, E. and Schabes, Y. (1997). Introduction. In Roche, E. and Schabes, Y., editors, *Finite−State Language Processing*, pages 1−65. MIT Press, Cambridge, Mass.

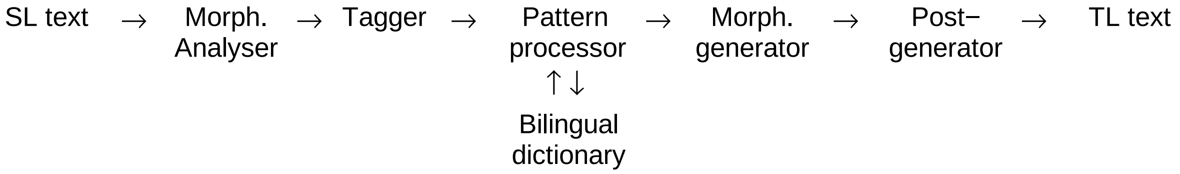

Figure 1: The linguistic modules in *interNOSTRUM* (deformatting and reformatting modules not depicted).

Table 1: Automatic generation of *interNOSTRUM*'s modules from linguistic data

LANGUAGE	LINGUISTIC DATA	GENERATION PROGRAM	interNOSTRUM MODULE
SL	morphological dictionary	Morphological analyser compiler	morphological analyser
SL	Morphologically analysed corpus	Tagger trainer	tagger
SL, TL	bilingual dictionary	Bilingual dictionary compiler	bilingual dictionary module

SL, TL	pattern processing rules	Pattern processing rule compiler	pattern processing module
TL	morphological dictionary	Morphological generator compiler	morphological generator
TL	apostrophe & hyphen rules	Postgenerator compiler	postgenerator