Open-source Portuguese–Spanish machine translation

C. Armentano-Oller[1], R.C. Carrasco[1,2], A.M. Corbí-Bellot[1], *M.L. Forcada*[1,2], M. Ginestí-Rosell[1], S. Ortiz-Rojas[2], J.A. Pérez-Ortiz[1,2], G. Ramírez-Sanchez[2], F. Sánchez-Martínez[1,2], M.A. Scalco[2]

[1]Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant (Spain)

[2]Prompsit Language Engineering, S.L., E-03690 St. Vicent del Raspeig (Spain).

PROPOR 2006 — Itatiaia, RJ, Brazil — May 15, 2006

Contents

1. The Apertium MT toolbox
   - Background
   - Rationale
   - Why open source?
   - The Apertium architecture
   - Modules
2. Linguistic data for the Portuguese–Spanish pair
   - Lexical data
   - Lexical disambiguation data
   - Structural transfer rules
   - Post-generation data
   - A quick evaluation
3. Concluding remarks

Background

Apertium is based on the technologies developed by the Transducens group at the Universitat d'Alacant during the development of two existing systems:

- **interNOSTRUM** (interNOSTRUM.com, Spanish–Catalan)
- **Tradutor Universia** (tradutor.universia.net, Spanish–Portuguese)

Rationale /1

To generate translations which are

- reasonably intelligible and
- easy to correct

between related languages such as Spanish (es), Catalan (ca), Portuguese (pt), etc.), one can just augment *word for word* translation with

- robust lexical processing (including multi-word units)
- lexical categorial disambiguation (part-of-speech tagging)
- local structural processing based on simple and well-formulated rules for frequent structural transformations (reordering, agreement)

Rationale /2

- It should be possible to generate the whole system from linguistic data (monolingual and bilingual dictionaries, grammar rules) specified in a declarative way.
- This information should be provided in an interoperable format $\Rightarrow$ XML. There are four basic file types (DTDs):
  - (language-independent) rules to treat text formats
  - specification of the part-of-speech tagger
  - morphological and bilingual dictionaries and dictionaries of orthographical transformation rules
  - structural transfer rules

Rationale /3

- It should be possible to have a single generic (language-independent) engine reading language-pair data (“separation of algorithms and data”)
- Language-pair data should be preprocessed so that the system is fast ($>$10,000 words per second) and compact; for example, lexical transformations are performed by minimized finite-state transducers (FSTs).

Why open source? /1

Reasons for the *open-source* development of Apertium:

- To give everyone free, unlimited access to machine-translation technologies.
- To establish a modular, documented, open platform for shallow-transfer machine translation and other human language processing tasks
- To favour the interchange and reuse of existing linguistic data.
- To make integration with other open-source technologies easier.

Why open source? /2

More reasons for open-source development of Apertium:

- To benefit from collaborative development
  - of the machine translation engine
  - of language-pair data for currently existing or new language pairs

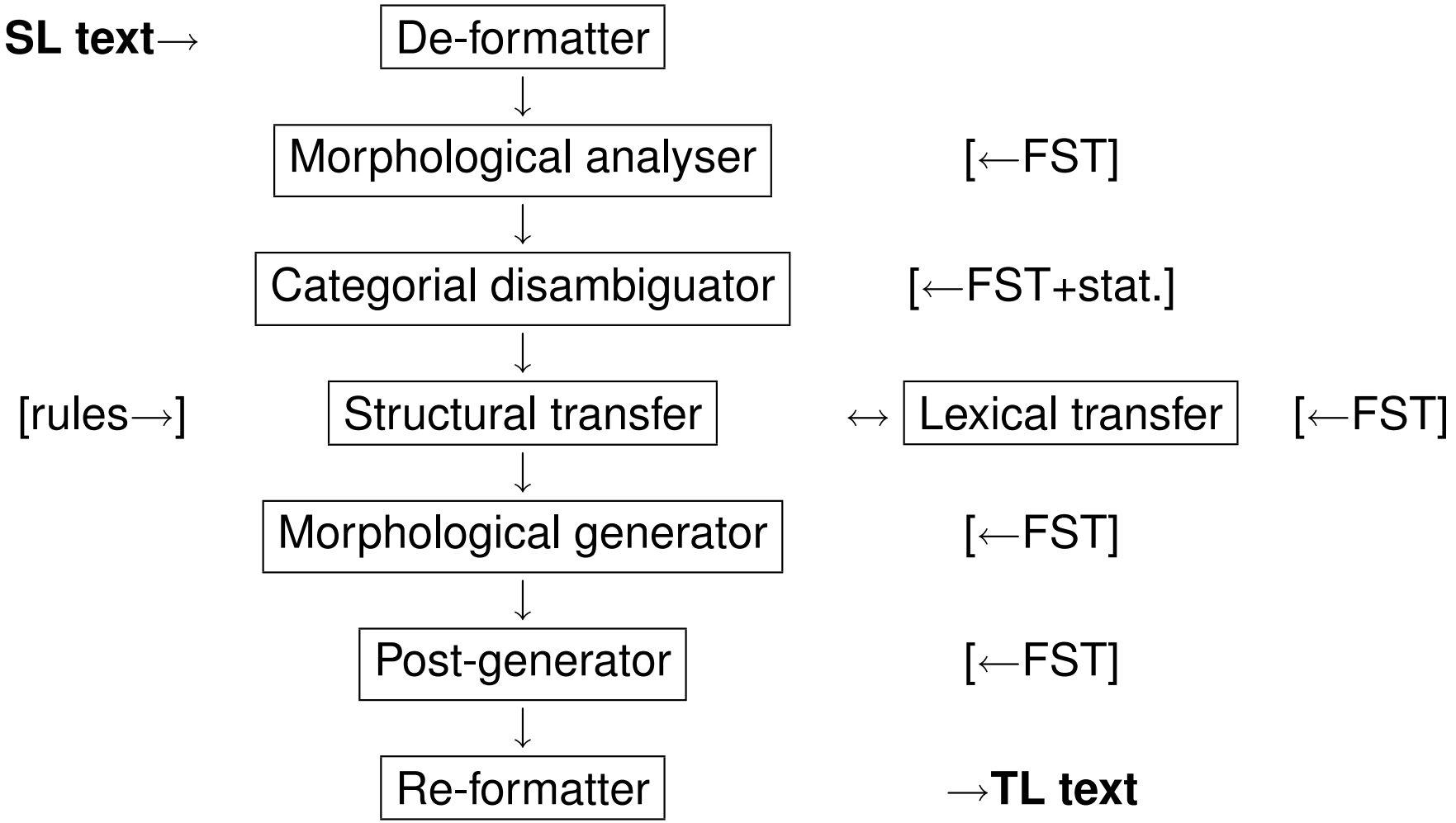
  from industries, and academia and minority-language support groups.
- To help shift MT business from an obsolescent *licence-centered* model to a *service-centered* model.
- To radically guarantee the reproducibility of our natural language processing research
- Because it does not make sense to use public funds to develop non-free, closed-source software.

The Apertium architecture/1

Apertium is an open-source machine translation toolbox (http://www.apertium.org) providing:

1. An open-source modular shallow-transfer machine translation **engine** with:
   - text format management
   - finite-state lexical processing
   - statistical lexical disambiguation
   - shallow transfer based on finite-state pattern matching
2. Open-source **linguistic data** in well-specified XML formats for a variety of language pairs (currently including Spanish–Catalan, Spanish–Galician and Spanish–Portuguese)
3. Open-source **compilers** to transform these linguistic data into a fast and compact form used by the engine

The Apertium architecture/2



The Apertium architecture/3

Communication between modules: text (*pipeline*).
Advantages:

- Simplifies diagnosis and debugging
- Allows the modification of data between two modules using, e.g., filters
- Makes it easy to insert alternative modules (interesting for research and development purposes)

De-formatter

- Separates text from format information
- Currently available for ISO-8859-1 plain text, HTML and RTF
- Based on finite-state techniques (`lex`)
- Generated (using a XSLT stylesheet) from an XML de-formatter specification file

Morphological analyser

- segments the source text in *surface forms* (SFs),
- assigns to each SF one or more *lexical forms* (LFs), each one with:
  - lemma
  - lexical category (part-of-speech)
  - morphological inflection information
- processes contractions (pt: *das*, es: *démonoslos*, etc.) and multi-word units which may be invariable (pt: *no entanto*) or variable (pt: *procurar pêlo em ovo*).
- reads finite-state transducers (letter transducers) generated from a morphological dictionary in XML (using a compiler)

Categorial disambiguator (part-of-speech tagger)

- picks one of the LFs corresponding to each ambiguous SF (about 30 % of them) according to context
- uses hidden Markov models and hand-written constraint rules
- is trained using representative corpora for the source language (manually disambiguated or not) or, recently, using statistical models for the TL.
- its behavior is completely specified by an XML archive

Structural transfer /1

- It is based on finite-state techniques (finite-state recognizers)
- The XML transfer-rule file is preprocessed for faster interpreting (compiling is also possible, but generates language-pair-dependent code which has to be compiled)
- Rules have a *pattern–action* form
- It detects LF patterns to be processed using a left-to-right, longest-match strategy
- It executes the actions associated to each pattern in the rule file to generate the corresponding LF pattern for the TL (Portuguese–Spanish examples later).

Lexical transfer module

- Reads each SL LF and generates the corresponding TL LF
- It reads finite-state transducers (letter transducers) generated from bilingual dictionaries in XML (using a compiler).
- It is invoked by the structural transfer module

Morphological generator

- Generates from each TL LF, a TL SF, after adequately inflecting it
- It reads finite-state transducers (letter transducers) generated from a morphological dictionary in XML (using a compiler)

Post-generator

- Performs some TL orthographical transformations, such as contractions (pt: *de* + *os* → *dos*; *dizer* + *o* → *dizê-lo*), inserting apostrophes (ca: *de* + *amics* → *d'amics*).
- It is based on finite-state transducers (letter transducers) generated from a post-generation rule dictionary (using a compiler)

Re-formatter

- Integrates format information (plain ISO-8859-1 text, HTML and RTF) into the translated text.
- It is based on finite-state techniques (`lex`)
- It is generated (using a XSLT stylesheet) from an XML de-formatter specification file

Linguistic data for the Portuguese–Spanish pair

- Data open for active development through the SourceForge CVS server (developers sought: contact `mlforcada@users.sourceforge.net`).
- Last `pt–es` language-pair data package released: `apertium-es-pt-0.9` [May 5, 2006]

Lexical data

- Lexical data are for *European* Portuguese, but currently recognize many Brazilian Portuguese forms in the pt→es direction
- The pt morphological dictionary contains 10,440 lemmas
- The es morphological dictionary contains 11,629 lemmas
- The es–pt bilingual dictionary contains 11,307 lemma–lemma correspondences

Lexical disambiguation data

- The *fine* inflection information delivered by the morphological analyser is grouped in a small set of *coarse* tags (pt: 128 tags, es: 78 tags).
- To simplify the lexical part of the HMM, words are grouped in ambiguity classes (pt: 459 classes, es: 260 classes)
- HMM trained from small hand-tagged corpora (pt: 29,214 words, es: 22,491 words) and retrained over larger untagged corpora (pt: 454,197 words, es: 520,091 words).
- Resulting HMM parameters provided in package (no need to retrain when installing)

Structural transfer rules/1

- About 90 structural transfer rules in either direction
- About 20 “generic” rules to ensure gender and number agreement in 20 noun-phrase patterns:

  um sinal vermelho (Portuguese, masc., “a red signal”) → *una señal roja* (Spanish, fem.)
- About 70 specific rules (next 3 slides)

Structural transfer rules/2

- Agreement of adjectives in sentences with the verb *ser* (“to be”): *O sinal é vermelho* (pt, masc., “The signal is red”) → *La señal es roja* (es, fem.).
- Verb tense selection: for example, *futuro do conjuntivo*:
  quando vieres (pt) [“when you come”] → *cuando vengas* (es)
  se vieres (pt) [“if you came”] → *si vinieras* (es)

Structural transfer rules/3

- Clitic pronoun rearrangements:
  enviou-me (pt) → *me envió* (es) [“he/she/it sent me”];
  para te dizer (pt) → *para decirte* (es) [“to tell you”], etc.
- Adding the preposition *a* in modal constructs: (*vai comprar* (pt) → *va a comprar* (es) [“is going to buy”]).
- Comparatives:
  - reordering (*mais dois carros* (pt) → *dos coches más* (es) [“two more cars”]) and
  - translating *do que* (pt) [“than”] as *que* (es) in *mais*. . . *do que*. . . constructs

Structural transfer rules/4

- Lexical rules: translating
  muito (pt) → *muy*/*mucho* (es) [“very”, “much”] or
  primeiro (pt) → *primer*/*primero* (es) [“first”].
- Progressive constructs: (pt) *estar a* + infinitive → (es) *estar* + gerund (*to be* + *-ing*).
- Syntactic changes: (pt) “gosto de cantar” (“I like to sing”) → (es) “me gusta cantar”.

Post-generation data

Rules are word-boundary string-to-string correspondences; regularities (*de* + *aquele/a/es/as* → *daquele/a/es/as*) are grouped in substring-to-substring paradigms

- pt: 54 rules, using 16 paradigms
- es: 26 rules, using 5 paradigms

A quick evaluation

A preliminary evaluation (5000 words) on version 0.9 (May 5, 2006).

- pt–es: 94.4 % coverage, 11,3 % error[1]
- es–pt: 94.5 % coverage, 4,7 % error
- Speed surpassing 5,000 words per second on a Pentium IV at 3 GHz.

[1] Number of 1-word edit operations (insertions, deletions, substitutions) per 100 words

Concluding remarks

- An open-source pt–es linguistic data package for the Apertium open-source shallow-transfer machine translation toolbox has been described.
- Very promising results are obtained with dictionaries containing 10,000 lemmas and with rule files containing less than 100 transfer rules.
- These results may improve dramatically when linguistic data grow (collaboration welcome!).
- Further improvements may come from enhancements to the Apertium architecture which is actively being developed.

Project funding

Partially funded by

- The Ministry of Industry, Tourism and Commerce of Spain (grants FIT-340101-2004-0003 and FIT-340001-2005-0002) funded the OpenTrad consortium (four universities, three enterprises) to develop the Apertium architecture.
- Funding from other ministries (grants TIC2000-1599-C02-01, TIC2003-08681-C02-01, HBP-2005-0018) is also acknowledged.
- The Spanish Ministry of Education and Science and the European Social Fund support Felipe Sánchez-Martínez (grant BES-2004-4711).