Apertium: Free/open-source rule-based machine translation

Mikel L. Forcada[1,2,3]

[1]Centre for Next Generation Localisation, Dublin City University, Dublin 9 (Ireland)
[2]Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, E-03071 Alacant (Spain)
[3]Prompsit Language Engineering, S.L., St. Francesc, 74, 1-L, E-03195 l'Altet (Spain)

Machine Translation Marathon, Dublin, Jan. 29, 2010

Contents

MT software/1

- MT is special: it strongly depends on *data*
  - *rule-based* MT (RBMT): dictionaries, rules
  - *corpus-based* MT (CBMT): sentence-aligned parallel text, monolingual corpora
- Three components in every MT system:
  - *Engine* (also *decoder*, *recombinator*...)
  - *Data* (linguistic data, corpora)
  - *Tools* to maintain these data and to convert them to the format used by the engine

MT software/2

Some reasons to use RBMT:

- CBMT requires massive amounts of sentence-aligned parallel text (a scarce resource for many language pairs).
- RBMT may use linguistic data elicited by speakers without access to existing machine-readable resources.
- RBMT is more transparent: errors are easier to diagnose and debug.

MT software/3 : commercial machine translation

- Most commercial MT systems are RBMT (but, for instance, LanguageWeaver, Google Translate are CBMT).
- They use proprietary technologies which are not disclosed (perceived as their main competitive advantage).
- For most users, only partial modification (*customization*) of linguistic data is allowed.

MT software/3: free/open-source machine translation

- For MT to be free/open-source (FOS), the *engine*, the *data* and the *tools* must all be free/open-source
- In the case of CBMT this means that corpora must also be free/open-source (hard to come by!)

Opportunities from free/open-source MT systems

- Even if reasonable-quality closed-source MT is available for a given language pair, the development and use of free/open-source MT systems provides *additional* opportunities:
  - Increases language **expertise** and **resources**
  - Increases **technological independence**

Increasing expertise and language resources

- When building a free/open-source MT system for a language pair, a variety of situations may occur:
  - Building linguistic data from scratch for an existing engine
  - Transforming existing linguistic data for one language pair into data for another language pair
  - Changing the engine to deal with new problems
- All of them involve building linguistic expertise and resources through
  - reflection about the languages involved
  - elicitation of linguistic (monolingual and bilingual) knowledge about them
  - subsequent encoding of this knowledge
- The free/open-source setting makes the newly created expertise and resources naturally available to the community.

Increasing technological independence

- Having a free/open-source engine, tools and data makes users of the involved languages less dependent on a single commercial, closed-source provider.
- This has an analogous effect, not only on machine translation, but also on other human language technologies.

Existing free/open-source rule-based MT systems

These are the three main FOS RBMT systems currently being actively developed:

- the Matxin MT system for Basque (`http://matxin.sf.net`),
- the OpenLogos MT system (`http://logos-os.dfki.de/`), and
- Apertium, which I will present here.

Matxin

- FOS MT system architecture for pair `es→eu` (`en→eu` being worked on).
- Uses a dependency parser for `es` based on Freeling and performs deep transfer; lexical transfer and generation use Apertium components.
- A branch uses Apertium components together with constraint grammar for analysis.
- Developed by group Ixa at Euskal Herriko Unibertsitatea and Elhuyar R&D both in the Basque Country.
- FOS software under the GPL license.

OpenLogos

- FOS version of historical system Logos (developed over 30 years).
- Language pairs: en↔de, en→fr, en→it, en→pt, en→es.
- Complex transfer, with semantics.
- Scarce documentation.
- Language data in Postgres data base form (no real “sources”)
- Multiple-licensed, but FOS under the GPL license.

Apertium: the inception

- October 2004: The Spanish Ministry of Industry, funds a consortium to build FOS MT for the languages of Spain:
  - Universities: EHU, UA, UPC, UVigo
  - Companies: Eleka, Elhuyar, Imaxin Software
- Project develops two systems:
  - Apertium (`es`↔`ca`, `es`↔`gl`)
  - Matxin (`es`→`eu`)

Technology/1

- Apertium not built from scratch.
- Complete FOS re-specification, rewriting and extension of closed-source systems built by Transducens at the UA:
  - **interNOSTRUM** (`interNOSTRUM.com`, `es↔ca`)
  - **Tradutor Universia** (`tradutor.universia.net`, `es↔pt`)
- Linguistic data for `es↔ca` and `es↔gl` built combining in-house resources with existing FOS data (e.g., in Freeling).

A conservative design? /1

Most of the design of Apertium is rather “conservative”:

- **No “rocket science”:** tested and established techniques and technologies: finite state transducers, finite-state pattern matching, hidden Markov models.
- **High-school linguistics:** representation based on well-known and widely-accepted linguistic concepts (morphology, parts of speech and just a little bit of syntax).

A conservative design? /2

- **Good-old 70's Unix style:** modularity achieved "the Unix way":
  - little programs "that do one thing and do it well" (McIlroy 1978)
  - "simple parts that are connected by clean interfaces" (Raymond 2004)
  - text, pipes & filters

  for easy diagnosis, extension, to build *frankensteins*, etc.

Development of language pairs as a driving force for innovation

Language-pair development (currently 21 stable pairs) has motivated changes in the Apertium platform:

- Apertium 1.0: designed to treat with closely-related language pairs (es↔ca, es↔pt, etc.)
- Apertium 2.0: three-stage structural transfer introduced to deal with less-related languages such as en↔ca
- Apertium 3.0: Unicode compliance to deal with any written language in the world
- multi-stage (>3) structural transfer for eo→en
- integration of VISL constraint grammar, motivated by
  - FOS grammars for no (nn, nb) and the Sámi languages
  - their utility to deal with the morphology of Celtic languages.

Build on top of word-for-word translation/1

To generate translations which are

- reasonably intelligible and
- easy to correct (*postedit*)

between related languages such as `es–ca`, `es–pt`, `nn–nb`, `ga–gd`, 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)

Build on top of word-for-word translation /2

For harder, not so related, language pairs:

- One should be able to build as much as possible on top of that simple model.
- It should be possible to generalize its concepts so that linguistic complexity is kept as low as possible.

Clear and effective separation of translation engine and language-pair data/1

- 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, i.e.,
  - (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

  should be provided in an interoperable format $\Rightarrow$ XML.

Clear and effective separation of translation engine and language-pair data/2

- 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).

Apertium as free/open-source software /1

Reasons for the development of Apertium as free/open-source software:

- To give everyone free, unlimited access to the best possible 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 free/open-source technologies easier.

Apertium as free/open-source software /2

More reasons for the development of Apertium as free/open-source software:

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

  from industries, academia and independent developers.
- To help shift MT business from the obsolescent *licence-centered* model to a *service-centered* model.
- To radically guarantee the *reproducibility* of machine translation and natural language processing research.
- Because public research investments must be made available to the public.

Reasons for the use of copyleft

What is *copyleft*?

- Obviously a play on the word *copyright*.
- Copyleft, when added to a free license, means that modifications have to be distributed with the same (copylefted) license.

Apertium chose *copylefted* free/open-source licences from the very beginning:

- To enable communities of programmers to build a machine translation *commons* or *pool* (Streiter et al. 2006), that is, a shared body of FOS machine translation software and data that stands stands a better chance of being preserved and extended. . .
- while allowing for many uses (including commercial uses).

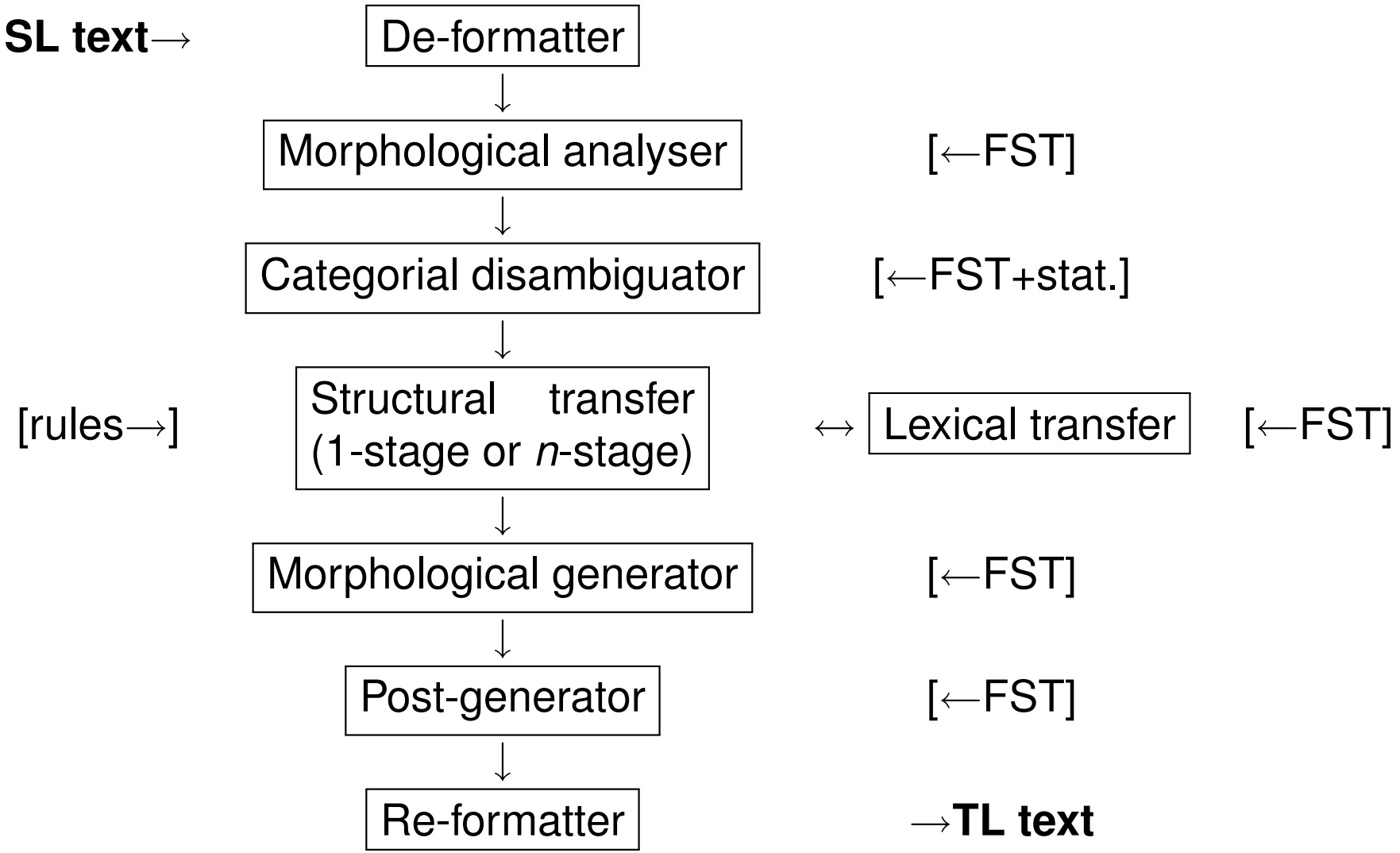
The license chosen was the GNU General Public License (GPL)

The Apertium platform

Apertium is a free/open-source machine translation platform (`http://www.apertium.org`) providing:

1. A free/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. Free/open-source **linguistic data** in well-specified XML formats for a variety of language pairs
3. Free/open-source tools: **compilers** to turn linguistic data into a fast and compact form used by the engine and software to learn disambiguation or structural transfer rules.

Architecture/1



Architecture/2

XML linguistic data are compiled for speed:

- Lexical information (SL and TL morphological dictionaries, SL–TL bilingual dictionaries, post-generation rules) → finite-state transducers (FST).
- Patterns identifying the left-hand side of structural transfer rules → finite-state pattern matchers
- Disambiguation rules and probabilities obtained from text corpora → hidden Markov models (HMM)
- etc.

The Apertium community/1

Not the ideal community development situation, but close. In addition to the original (funded) developers, a community (instigated by Francis Tyers) formed around the platform.

- More than 100 developers in `sourceforge.net/projects/apertium/`, many outside the original group (thank you all!)
- Code updated very frequently: hundreds of monthly SVN commits
- A collectively-maintained *wiki* shows the current development and tips for people building new language pairs or code.

The Apertium community/2

- Externally developed tools and code:
  - a graphical user interface `apertium-tolk`, and the related diagnostic tool `apertium-view` and `apertium-view`
  - plugins for OpenOffice.org, the Pidgin (previously Gaim) messaging program, for the Wordpress content management system, the Virtaal translation software, the Jubler film-subtitling application, etc.
  - A standalone film subtitling application (`apertium-subtitles`)
  - Dictionaries adapted to mobile phones and handhelds (`tinylex`)
  - Windows ports.
- Many people gather and interact in the `#apertium` IRC channel (at `freenode.net`).
- Stable packages ported to Debian GNU/Linux (and therefore to Ubuntu and gNewSense).

Research/1

- Apertium is also a MT research platform.
- New code (`apertium-tagger-training-tools`, `apertium-transfer-tools`) or language-pair data have often been released simultaneously to research publications.
- The research undertaken has even produced a PhD thesis (Felipe Sánchez-Martínez 2008) and four master's theses (Gema Ramírez-Sánchez, Carme Armentano-Oller, Francis M. Tyers, Ángel Seoane).
- A survey of published research may be found in the paper.
- Apertium has also been used to obtain resources for other MT systems.

Research/2

Access to FOS software like Apertium

- guarantees the reproducibility of all of the above experiments
- “lowers the bar for entry to your project for new colleagues” (Pedersen 2008: “Empiricism is not a matter of faith”, recommended reading!)

Research/3

Together with other FOS machine translation software, such as

- the Giza++ statistical aligner,
- the Moses statistical MT engine,
- the IRSTLM language-model toolkit,
- the Cunei example-based MT platform,
- the Anymalign aligner,
- the Matxin MT system for Basque, and
- the OpenLogos MT system,

Apertium contributes to the reproducibility and the advancement of MT research and experiments.

Business with Apertium

Companies in the initial consortium sell services based on Apertium:

- Eleka Ingeniaritza Linguistikoa
- imaxin|Software

Prompsit Language Engineering, started in 2006:

- works almost exclusively on Apertium
- currently one of the main developers of the platform

Services:

- installing and supporting translation servers
- maintaining and extending language-pair data for a particular application
- integrating Apertium in multilingual documentation management systems

Recent developments: the 2009 Google Summer of Code

Apertium was selected to participate as a mentoring organisation in the 2009 Google Summer of Code. Succesful projects:

- two new language pairs: `nn`↔`nb` and `sv`↔`da`
- a morphological analyser for `bn`
- an improved part-of-speech tagger
- a web-service infrastructure
- porting of the lexical component to Java
- hybridising Apertium with other systems

Recent developments: ongoing work

- Universidá d'Uviéu: `es↔ast`
- University of Reykjavík: `is→en`
- Universitat d'Alacant and Prompsit: `es↔it`
- University of Tromsø: `sme→nob` `sme↔smj`

Lots of work ahead: known limitations

- No successful, general-purpose lexical selection for polysemic words
- No deep (parse-tree-based) structural transfer, needed for syntactically divergent language pairs
- Current lexical processing not adequate for agglutinative languages or languages with non-catenative morphology.
- The representation of morphological inflection is still too low-level.
- No support to segment long compound words (`de`: *Kontaktlinsenverträglichkeitstest*)
- Apertium is a *transfer* system: generating a new pair involves the creation of explicit bilingual resources. `apertium-dixtools` helps build pair *A*–*B* from *A*–*C* and *C*–*B*, but task is far from trivial.

Funding

Apertium has been funded by

- The Ministry of Industry, Tourism and Commerce of Spain (also, the Ministries of Education and Science and of Science and Technology of Spain)
- The Secretariat for Technology and the Information Society of the Government of Catalonia
- The Ministry of Foreign Affairs of Romania
- The Universitat d'Alacant
- The Ofis ar Brezhoneg (Breton Language Board)
- Google (Google Summer of Code 2009) scholarships
- Companies: Prompsit Language Engineering, ABC Enciklopedioj, Eleka Ingeniartiza Linguistikoa, imaxin|software, etc.

License

This work may be distributed under the terms of

- the Creative Commons Attribution–Share Alike license: `http://creativecommons.org/licenses/by-sa/3.0/`
- the GNU GPL v. 3.0 License: `http://www.gnu.org/licenses/gpl.html`

Dual license! E-mail me to get the sources: `mlf@ua.es`