The Apertium machine translation platform: Five years on

Mikel L. Forcada*
Dublin City University
Glasnevin, Dublin 9
Ireland
mforcada@computing.dcu.ie

Francis M. Tyers
Dept. Lleng. i Sist. Inform.
Universitat d'Alacant
E-03071 Alacant, Spain
ftyers@dlsi.ua.es

Gema Ramírez-Sánchez,
Prompsit Lang. Engineering,
Av. St. Francesc, 74, 1r-L,
E-03195 L'Altet, Spain
gramirez@prompsit.com

Abstract

This paper describes Apertium: a free/open-source machine translation platform (engine, toolbox and data), its history, its philosophy of design, its technology, the community of developers, the research and business based on it, and its prospects and challenges, now that it is five years old.

1 Introduction

This paper describes the Apertium free/open-source (FOS) machine translation (MT) platform: its history (Sec. 2), the philosophy behind its design (Sec. 3) its technology (Sec. 4), its community of users and developers (Sec. 5), research and business based on it (Sections 6 and 7), and its prospects and limitations (9).

2 History

2.1 The inception

One could say that the history of Apertium starts in April 2004, when Mikel L. Forcada sends around an e-mail message to a number of human language technology research groups in Spain on the possibility of lobbying the government agencies involved so that they fund the effort to build an FOS MT system for the languages of Spain.[1] Later on, in July 2004, the Spanish Ministry of Industry, Tourism and Commerce launched a call to fund consortia to develop linguistic technology for the languages of Spain. Some of the groups that answered yes to the above e-mail formed a consortium [2] and got funding for the development of two different MT systems: Matxin,[3] for Spanish (es) to Basque (eu), and Apertium, to translate between es and Catalan (ca) and es and Galician (gl).

The MT engine and tools in Apertium were not built from scratch, but are rather the result of a complete FOS rewriting and extension of two existing MT systems developed by the Transducens group at the Universitat d'Alacant, namely interNOSTRUM[4] (Canals-Marote et al., 2001) (es–ca) and Traductor Universia[5] (Garrido-Alenda et al., 2004) (es–Portuguese (pt)),to provide a platform to build MT systems for related languages. Linguistic data for the initial language pairs were built combining in-house resources with FOS data such as the ones present in Freeling[6] (Carreras et al., 2004).

[*] Mikel L. Forcada is supported by Spanish MEC project TIN2009-14009-C02-01. Permanent address: Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, E-03071 Alacant

[1] In addition to Spanish, official in the whole of Spain, four languages are co-oficial in some areas: Basque, Galician, Catalan (also called Valencian), and Occitan (Aranese). Other languages such as Asturian or Aragonese have a more limited legal status.

[2] This consortium, which involved 4 universities (Universitat d'Alacant, Universitat Politècnica de Catalunya, Universidade de Vigo and Euskal Herriko Unibertsitatea) 2 companies (Eleka Ingeniaritza Linguistikoa, imaxin|software) and 1 Foundation (Elhuyar) adopted the name Opentrad during the project. Currently this name is used as a trademark by some of the companies in the original consortium to commercialize machine translation services based on Apertium (http://www.opentrad.com).

[3] http://matxin.sf.net

[4] http://www.internostrum.com

[5] http://traductor.universia.net

[6] http://www.lsi.upc.edu/~nlp/freeling/

2.2 Technology

The initial shallow-transfer strategy used in the Apertium platform, briefly described in detail in sec. 4, was initially designed to treat pairs of closely related languages. The platform has been extended since version 2 to treat less closely related pairs, such as `es`–French (`fr`) (by enhancing the structural transfer); the results are far from satisfactory for some applications but we believe the possibilities offered by Apertium have not been thoroughly tested yet. To be able to deal with any written language in the world, Apertium was made Unicode-compliant as of version 3.

2.3 A conservative design?

However, most of the design of Apertium is very conservative in many respects:

- **No rocket science:** To achieve fast translation speeds in the range of 10,000 words a second on regular desktop computers (no need for Google-sized server farms!), Apertium uses tested and established techniques and technologies: finite-state transducers (FSTs) for lexical processing, hidden Markov models for part-of-speech tagging, and multi-stage finite-state chunking for structural transfer.
- **High-school linguistics:** To lower the bar for entry of developers, the representation of linguistic data is kept as simple as possible by basing it on well-known and widely-accepted linguistic concepts (morphology, parts of speech and just a little bit of syntax).
- **Good-old 70's Unix style:** Apertium achieves modularity by following the Unix philosophy: it is made of little programs "that do one thing and do it well" (McIlroy et al., 1978), "simple parts that are connected by clean interfaces" (Raymond, 2004) and talk *text* to each other through pipes. Such a structure is very amenable to multiprocessing and takes full advantage of new hardware trends. It also facilitates diagnosis and insertion of new modules (e.g. constraint grammars (Karlsson, 1995) in the Welsh–English pair (Tyers and Donnelly, 2009)), and even build *frankensteins*,[7] if need arise.

2.4 Development of language pairs as a driving force for innovation

Aside from the original two language pairs (`es`–`ca` and `es`–`gl`), a number of pairs have been built outside the initial consortium. These have been developed: with academia, as research projects by students (e.g. `fr`–`ca`, `pt`–`ca` and Welsh (`cy`)–`en`) and research groups (`en`–`ca`, `en`–`es`, `es`–Romanian (`ro`)), by companies involved in development of Apertium (e.g. `fr`–`es`, `pt`–`gl`, Occitan (`oc`)–`ca`) and in within the Apertium community (`en`–Esperanto (`eo`), Norwegian Nynorsk (`nn`)–Norwegian Bokmål (`nb`), Swedish (`sv`)–Danish (`da`), Breton (`br`)–`fr`).

Figure 1 shows the development of the number of language pairs by year, from 2005 when the first pairs were published to the present day.

Language-pair development has also motivated changes in the platform: three-level transfer was introduced to deal with `en`–`ca`, and multi-level transfer for `eo`–`en`. The integration of the VISL constraint grammar system[8] was largely motivated by FOS grammars for `nb` and the Sámi languages and their utility to deal with the morphology of Celtic languages.

3 The Apertium philosophy

Build on top of word-for-word translation: One of the basic design principles of Apertium was inherited from the previous works: to generate translations which are, on one hand, reasonably intelligible in their raw form and on the other hand, easy to correct (*postedit*) into publishable translations, between related languages , one can just augment *word for word* translation with robust lexical processing (including multi-word units), lexical categorial disambiguation (part-of-speech tagging), and local structural processing based on simple and well-formulated rules for frequent structural transformations (reordering, agreement).

It should also be possible to generalize the main concepts of this model to deal with harder, not-so-related language pairs, so that linguistic complexity is kept as low as possible.

[7] Systems made of Apertium modules and modules from other systems.

[8] `http://vislcg.sourceforge.net/`

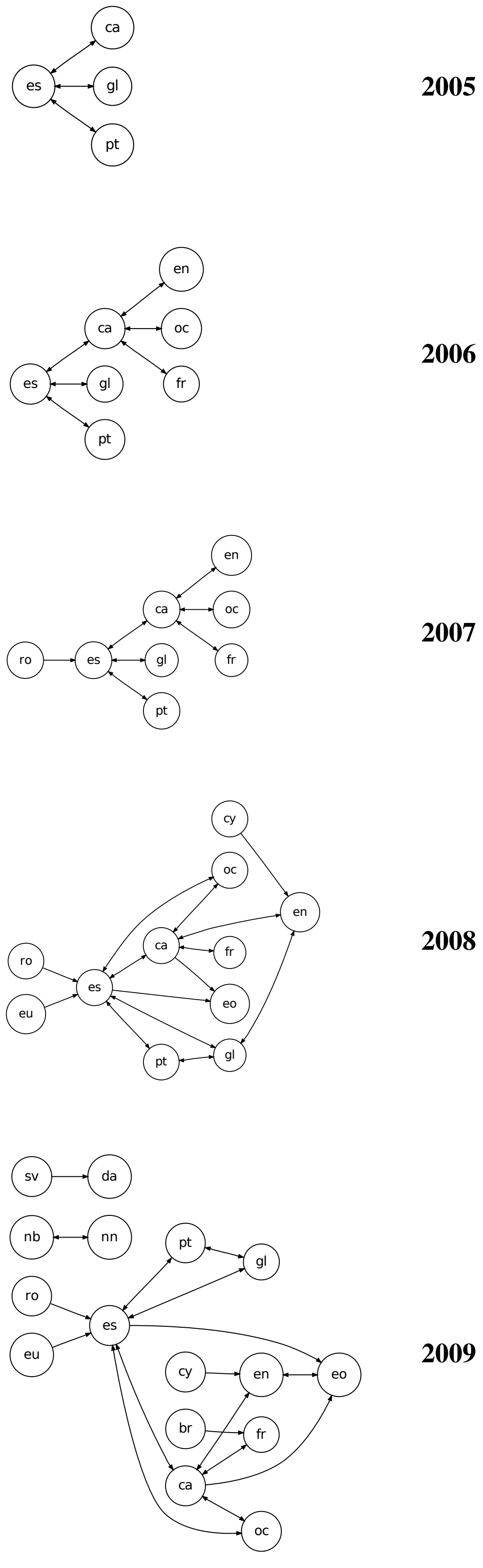

Figure 1: Language pair growth by year: 2005–2009

Clear and effective separation of translation engine and language-pair data: From the very beginning, the objective was that it should be possible to build a complete system for a specific language pair just by creating linguistic data specified in a declarative way. This information, i.e., language-independent rules to treat different document formats, the specification of the part-of-speech tagger, morphological and bilingual dictionaries, dictionaries of orthographical transformation rules, and structural transfer rules, should be provided in an interoperable format. This naturally led us to choose formats based on XML. Therefore, it was thought to be crucial to have a single generic (language-independent) engine reading language-pair data (usually referred to as "separation of algorithms and data").

The human-editable version of the language-pair data in XML should be preprocessed so that the system is fast enough and compact; for example, lexical transformations are performed by minimized FSTs.

Apertium as FOS software: After five years of experience in MT, reflection in the Transducens group at the Universitat d'Alacant settled around a FOS solution, for a wide variety of reasons:

- To give everyone free access to the best possible machine-translation technologies.
- To establish a modular, documented, open platform for shallow-transfer MT and other human language processing tasks.
- To favour the sharing and reuse of existing linguistic data and to make integration with other FOS technologies easier.
- To benefit from collaborative development of both the engine and language-pair data for existing or new language pairs, from industry, academia and independent developers.
- To help shift MT business from the obsolescent and vulnerable *licence-centred* model to a *service-centred* model (see sec. 7).
- To radically guarantee the *reproducibility* of experimental research on MT.
- Because the results of publically funded research must be made available to the public.

Reasons for the use of *copyleft*: The word *copyleft*, a play on the word *copyright*, when added to a free licence, means that modifications have to be distributed with the same (copylefted) licence. When it was time to choose a FOS licence for Apertium, we immediately settled for *copylefted* licences, because copyleft secures the existence of a commons of MT technology. Now all of Apertium is licenced under the GNU General Public Licence or GPL.[9] Copyleft protects the Apertium commons from private aappropriation (incorporation into non-free software), ensures that all new developments would be also FOS, and enables programmers to build a shared body of MT resources while allowing for commercial uses (see sec. 7).

4 Technology

A very brief description of Apertium will be given here. Turn to existing descriptions (such as (Armentano-Oller et al., 2006)) for details.

The Apertium platform provides: (a) A FOS modular shallow-transfer MT *engine* with text format management, finite-state lexical processing, statistical lexical disambiguation, and shallow structural transfer based on finite-state pattern matching; (b) FOS *linguistic data* in well-specified XML formats for a wide variety of language pairs; and (c) FOS tools such as *compilers* to turn linguistic data into a fast and compact form used by the engine and software to learn disambiguation or structural transfer rules.

The Apertium engine is a pipeline or assembly line consisting of the following stages or modules:

- A *deformatter* which encapsulates the format information in the input document.
- A *morphological analyser* which segments the text in surface forms ("words") and delivers, for each surface form, one or more *lexical forms* consisting of *lemma*, *lexical category* and morphological inflection information. It reads a FST generated from a source-language (SL) morphological dictionary in XML.
- A *part-of-speech tagger* which chooses, using a first-order hidden Markov model (HMM), the most likely lexical form corresponding to an ambiguous surface form, as trained using a corpus and a tagger definition file in XML.
- A *lexical transfer* module which reads each SL lexical form and delivers the corresponding target-language (TL) lexical form by looking it up in a bilingual dictionary in XML using a FST generated from it.
- A *structural transfer*, generally consisting of three sub-modules (some language pairs use only one module and some others more than three):
  - A *chunker* which, after invoking lexical transfer, performs local syntactic operations and segments the sequence of lexical units into chunks. A chunk is defined as a fixed-length sequence of lexical categories that corresponds to some syntactic feature such as a noun phrase or a prepositional phrase.
  - An *interchunk* module which performs more global operations with the chunks and between them.
  - A *postchunk* module which performs finishing operations on each chunk.

  Each of the modules reads rules from files written in XML.
- A *morphological generator* which delivers a TL surface form for each TL lexical form, by suitably inflecting it. It reads a FST generated from a TL morphological dictionary in XML.
- A *post-generator* which performs orthographic operations such as contractions (e.g. `es` *del* = *de* + *el*) and apostrophations (e.g. `ca` *l'institut* = *el* + *institut*), using a FST generated from a rule file written in XML.
- A *reformatter* which de-encapsulates any format information.

5 The Apertium community

In addition to the original (publically-funded) developers, an active community of more than 100 developers,[10] most of them from outside the original group, has formed around the platform. Code, especially language-pair data, is updated very frequently: hundreds of monthly commits are made

[9] `http://www.fsf.org/copyleft/gpl.html`

[10] As registered in `http://www.sourceforge.net/projects/apertium/`

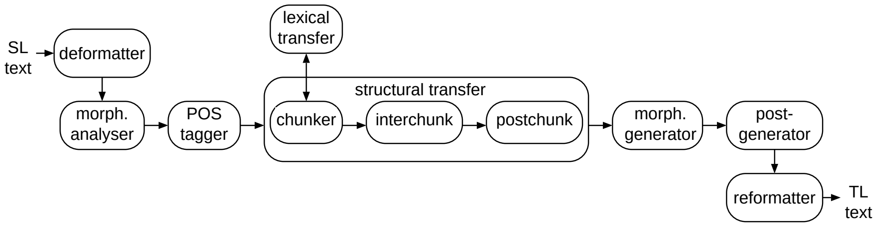

Figure 2: Modular architecture of the Apertium FOS MT platform.

to the project's software configuration management (SCM) system. A collectively-maintained *wiki*[11] shows the current development and gives tips to build new language pairs or code. Developers and users gather and interact in the `#apertium` IRC channel (at `irc.freenode.net`).

The strength of the Apertium community may also be measured by the large number of externally developed tools and code that add functionalities to Apertium, which include: a graphical user interface `apertium-tolk` and the related diagnostic tools `apertium-view` and `apertium-viewer`; plugins for OpenOffice.org[12], Wordpress[13], the Virtaal translation software,[14] and the Jubler film subtitling application[15]; a stand-alone film subtitling application (`apertium-subtitles`), and dictionary lookup tools for mobile phones and handhelds (`tinylex`)[16] All of the stable Apertium packages have been packaged for Debian GNU/Linux[17] and as a result, Apertium is part of one of the most popular GNU/Linux distributions, Ubuntu.[18]

6 Research

Apertium is also a MT research platform. New code or data have often been released simultaneously to research publications. The research undertaken has even produced a PhD thesis (Sánchez-Martínez, 2008) and four master's theses.[19] Here is a survey of published research:

- Several papers have describe the platform (Carme Armentano-Oller et al., 2005; Corbí-Bellot et al., 2005; Armentano-Oller et al., 2006; Ramírez-Sánchez et al., 2006; Armentano-Oller et al., 2007).
- Sánchez-Martínez et al. (2005, 2006, 2008) have explored use of TL information to train the SL part-of-speech disambiguator (package `apertium-tagger-training-tools`).
- Sánchez-Martínez and Forcada (2007, 2009) have studied the use of statistical methods to infer structural transfer rules from relatively small, sentence-aligned bilingual texts (package `apertium-transfer-tools`).
- A number of papers describe the creation of data for new Apertium language pairs, using a variety of approaches, including the reuse of existing FOS resources: `es–pt` (Armentano-Oller et al., 2006), `ca–pt` (Armentano-Oller and Forcada, 2008), `eu–es` (Ginestí-Rosell et al., 2009), two Sámi languages (Tyers et al., 2009), and `cy–en` (Tyers and Donnelly, 2009).
- Apertium resources have also been used to infer bilingual resources (dictionaries, rules) for other MT systems (Caseli and Nunes, 2006; Caseli et al., 2006), to assist training in statistical MT (Tyers, 2009), or in another transfer architecture (Alegria et al., 2005).

Access to FOS software (FOSS) like Apertium guarantees the reproducibility of all of the above experiments, and "lowers the bar for entry to your project for new colleagues" (Pedersen, 2008).

7 Business

The companies involved in the project where the development of Apertium started[20] offer nowa-

[11] `http://wiki.apertium.org`

[12] `http://www.openoffice.org`

[13] `http://wordpress.com/`

[14] `http://translate.sf.net/wiki/virtaal`

[15] `http://www.jubler.org`

[16] `http://www.tinylex.com`

[17] `http://www.debian.org`

[18] `http://www.ubuntu.com`, and its strictly-free alternative gNewSense, `http://gnewsense.org`.

[19] Those of Carme Armentano-Oller, Gema Ramírez-Sánchez, Francis M. Tyers, and Ángel Seoane, all at the Universitat d'Alacant.

[20] See footnote in p. 1

days services based on Apertium. Also Prompsit Language Engineering, started in 2006, works almost exclusively on Apertium and is currently one of the main developers of the platform.

Companies holding the copyright of a certain component of Apertium could in principle also sell non-free/closed-source software (NFCSS) based on it following the usual licence-centred model in the software industry. However, most of the business around Apertium follows a different model: they offer services around the Apertium platform allowed by its GPL licence. Services that maybe offered include installing and supporting translation servers; maintaining, adapting and extending linguistic data for a particular purpose ; building new language pairs; integrating MT systems in multilingual documentation management systems, or developing new tools around Apertium.[21]

There are indeed advantages for businesses using Apertium as a FOS MT solution, e.g.:

- Customers may prefer an FOS MT solution to NFCSS. In the new setting, vendors are not providers to whom they have a technological dependency, but are instead technological partners, since customers may feel free to shop around for services around the FOS system and hire any other company offering them.
- Companies get a positive social image by developing FOSS: they are not only offering a better service, but also benefiting the whole community, which may in turn choose them among other options in the market.
- Contributions to the FOS project coming from the community around it benefit companies so that they can offer better services.

The new business model is not free of vulnerabilities as, for instance, many companies may offer the same services on the same platform. However, companies heavily involved in the development of the platform may acquire a deeper knowhow that gives them an edge over their competitors.

8 Recent developments

Apertium was selected to participate as a mentoring organisation in the 2009 Google Summer of Code.[22] Of the successful projects, two new language pairs have been created (nn–nb and sv–da), a morphological analyser for Bengali has been created, and work has been done to improve the part-of-speech tagger and to create a web-service infrastructure for Apertium. Along with this, one project ported the finite-state toolkit lttoolbox to Java to facilitate reuse in applications using that language, and another worked on hybridising Apertium with other MT systems. Other projects are also ongoing. Teams at the Universidá d'Uviéu and the University of Reykjavík are working on language pairs for es–Asturian (ast) and Icelandic (is)–en respectively, and es–Italian (it) is also being worked on.

9 Lots of work ahead

9.1 Known limitations

The Apertium platform still shows a number of important limitations that have to be tackled to make it more apt to deal with all kinds of languages. Here are some of them:

- Polysemic SL words may have more than one TL equivalent. Apertium bilingual dictionaries give only one TL lemma for each SL. No successful, efficient, general-purpose lexical selection module has been implemented yet.
- The structural transfer component in Apertium does not rely on a real tree-like parse of the whole sentence, but rather on one or more levels of *chunking*. A more powerful structural transfer is planned but work has yet to start.
- The current design of morphological analysis and generation make it hard to write morphological dictionaries for agglutinative languages such as eu, Sámi or Inuktikut (iu). Also, their design is too geared toward suffix or prefix morphology, what makes it hard

[21] Examples of commercial successes: the integration of Apertium to add gl to the online edition of a newspaper originally published only in es, http://lavozdegalicia.es/ (imaxin|software, Universidade de Vigo), and the Catalan government's official es–oc and ca–oc translator: http://traductor.gencat.cat/ (Taller Digital, Prompsit).

[22] http://code.google.com/soc/

to treat, for instance, languages with non-catenative morphology, such as Arabic (ar).

- The management of inflection paradigms is still not powerful enough to represent all relevant regularities. Various ad-hoc attempts at using *metadix* formats that are then converted to the standard XML *dix* format of Apertium are found in some packages, but a general scheme is yet to be defined.
- Many languages, such as is write many compounds as single words and do so very productively. Apertium does not have a general mechanism to segment compounds into lexical units.
- The Apertium architecture is a transfer architecture. Generating a new pair involves the creation of explicit bilingual resources. Tools like apertium-dixtools can alleviate the task of building data for language pair A–B when data for A–C and C–B are available (Armentano-Oller and Forcada (2008)), but the task remains far from trivial.

Acknowledgements: Apertium has been funded by the Ministries of Industry, Tourism and Commerce, of Education and Science, and of Science and Technology of Spain, the Government of Catalonia, the Ministry of Foreign Affairs of Romania, the Universitat d'Alacant, Ofis ar Brezhoneg and Google Summer of Code. Many companies have also invested in it: Prompsit Language Engineering, ABC Enciklopedioj, Eleka Ingeniaritza Linguistikoa, imaxin|software, etc. It is impossible to name all of the independent developers that have made substantial contributions, so we will not even try. Browsing the project's SCM[23] gives an idea.

References

Alegria, I., Diaz de Ilarraza, A., Labaka, G., Lersundi, M., Mayor, A., Sarasola, K., Forcada, M. L., Ortiz-Rojas, S., and Padró, L. (2005). An open architecture for transfer-based mt between spanish and basque. In *OSMaTran, A workshop at MT Summit X (Phuket, Thailand, September 12–16, 2005)*, pages 12–16.

Armentano-Oller, C., Carrasco, R. C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., and Scalco, M. A. (2006). Open-source Portuguese–Spanish machine translation. In Vieira, R., Quaresma, P., Nunes, M., Mamede, N., Oliveira, C., and Dias, M., editors, *Computational Processing of the Portuguese Language, Proc. PROPOR 2006*, volume 3960 of *LNCS*, pages 50–59. Springer-Verlag.

Armentano-Oller, C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Montava Belda, M. A., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., and Sánchez-Martínez, F. (2007). Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática. In Rodríguez Galván, J. R. and Palomo Duarte, M., editors, *Proc. of the FLOSS International Conf. 2007*, pages 5–20. Servicio de Publicaciones de la Universidad de Cadiz.

Armentano-Oller, C. and Forcada, M. (2008). Reutilización de datos linguısticos para la creacion de un sistema de traduccion automatica para un nuevo par de lenguas. *PLN*, 41:243–250.

Canals-Marote, R., Esteve-Guillen, A., Garrido-Alenda, A., Guardiola-Savall, M., Iturraspe-Bellver, A., Montserrat-Buendia, S., Ortiz-Rojas, S., Pastor-Pina, H., Perez-Antón, P., and Forcada, M. (2001). The Spanish–Catalan machine translation system interNOSTRUM. In *Proc. of MT Summit VIII.* Santiago de Compostela, Spain, 18–22 July 2001.

Carme Armentano-Oller, C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Bonev, B., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., and Sánchez-Martínez, F. (2005). An open-source shallow-transfer machine translation toolbox: consequences of its release and availability. In *OSMaTran, A workshop at MT Summit X*, pages 23–30.

Carreras, X., Chao, I., Padro, L., and Padro, M. (2004). Freeling: An open-source suite of language analyzers. In *Proc. of the 4th LREC*, volume 4.

[23] http://apertium.svn.sf.net/viewvc/apertium/

Caseli, H., Nunes, M., and Forcada, M. (2006). Automatic induction of bilingual resources from aligned parallel corpora: application to shallow-transfer machine translation. *MT*, 20(4):227–245.

Caseli, H. M. and Nunes, M. G. V. (2006). Automatic transfer rule induction from parallel corpora. In *Proc. of the 3rd Workshop on MSc dissertations and PhD theses in AI (WTDIA) - International Joint Conf. IBERAMIA/SBIA/SBRN 2006*, pages 1–10.

Corbí-Bellot, A. M., Forcada, M. L., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Alegria, I., Mayor, A., and Sarasola, K. (2005). An open-source shallow-transfer machine translation engine for the romance languages of spain. In *Proc. of the Tenth Conf. of the European Association for MT*, pages 79–86.

Garrido-Alenda, A., Gilabert Zarco, P., Pérez-Ortiz, J. A., Pertusa-Ibáñez, A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M. A., and Forcada, M. L. (2004). Shallow parsing for Portuguese–Spanish machine translation. In Branco, A., Mendes, A., and Ribeiro, R., editors, *Language technology for Portuguese: shallow processing tools and resources*, pages 135–144. Edições Colibri, Lisboa.

Ginestí-Rosell, M., Ramírez-Sánchez, G., Ortiz-Rojas, S., Tyers, F. M., and Forcada, M. L. (2009). Development of a free Basque to Spanish machine translation system. *PLN*, (43):187–195.

Karlsson, F. (1995). *Constraint Grammar: a language-independent system for parsing unrestricted text*. Walter de Gruyter.

McIlroy, M., Pinson, E., and Tague, B. (1978). Unix time-sharing system forward. *The Bell System Technical J.*, 57(6 part 2):1902.

Pedersen, T. (2008). Empiricism is not a matter of faith. *Comp. Ling.*, 34(3):465–470.

Ramírez-Sánchez, G., Sánchez-Martínez, F., Ortiz-Rojas, S., Pérez-Ortiz, J. A., and Forcada, M. L. (2006). Opentrad Apertium open-source machine translation system: an opportunity for business and research. In *Proc. of Translating and the Computer 28 Conf.*

Raymond, E. S. (2004). *The Art of Unix Programming*. Addison-Wesley.

Sánchez-Martínez, F. (2008). *Using unsupervised corpus-based methods to build rule-based machine translation systems*. PhD thesis, Universitat d'Alacant.

Sánchez-Martínez, F. and Forcada, M. L. (2007). Automatic induction of shallow-transfer rules for open-source machine translation. In Way, A. and Gawronska, B., editors, *Proc. of the 11th Conf. of TMI (TMI 2007)*, volume 1, pages 181–190. Skövde University Studies in Informatics.

Sánchez-Martínez, F. and Forcada, M. L. (2009). Inferring shallow-transfer machine translation rules from small parallel corpora. *J. of AI Research*, 34:605–635.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2005). Target-language-driven agglomerative part-of-speech tag clustering for machine translation. In *Proc. of the International Conf. RANLP - 2005 (Recent Advances in NLP)*, pages 471–477.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2006). Speeding up target-language driven part-of-speech tagger training for machine translation. *LNCS*, 4293:844–854.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2008). Using target-language information to train part-of-speech taggers for machine translation. *MT*, 22(1-2):29–66.

Tyers, F. and Donnelly, K. (2009). apertium-cy–a collaboratively-developed free RBMT system for Welsh to English. *Prague Bull. of Math. Ling.*, 91:57–66.

Tyers, F. M. (2009). Rule-based augmentation of training data in breton–french statistical machine translation. In *Proc. of the 13th Annual Conf. of the European Association of MT, EAMT09*, pages 213–218.

Tyers, F. M., Wiechetek, L., and Trosterud, T. (2009). Developing prototypes for machine translation between two Sámi languages. In *Proc. of the 13th Annual Conf. of the EAMT, EAMT09*, pages 120–128.