Free/open-source resources in the Apertium platform for machine translation research and development

Francis M. Tyers, Felipe Sánchez-Martínez, Sergio Ortiz-Rojas,
Mikel L. Forcada

Abstract

This paper describes the resources available in the Apertium platform, a free/open-source framework for creating rule-based machine translation systems. Resources within the platform take the form of finite-state morphologies for morphological analysis and generation, bilingual transfer lexica, probabilistic part-of-speech taggers and transfer rule files, all in standardised formats. These resources are described and some examples are given of their reuse and recycling in combination with other machine translation systems.

1. Introduction

Apertium (`http://www.apertium.org`) is a free/open-source (FOS) platform for creating rule-based machine translation systems (Forcada, Tyers, and Ramírez-Sánchez, 2009). There are currently stable data for 21 language pairs available within the platform. Resources within the platform take the form of finite-state morphologies for morphological analysis and generation, bilingual transfer lexica, probabilistic part-of-speech taggers and transfer rule files, all in standardised formats. These resources are described and some examples are given of their reuse and recycling in combination with other machine translation systems.

This article is organised as follows: section 2 describes the Apertium engine; section 3 describes the current status of the resources available in the platform; section 4 gives some details of ways these resources can be re-used within other machine translation systems, finally section 5 gives some directions of future work and discussion.
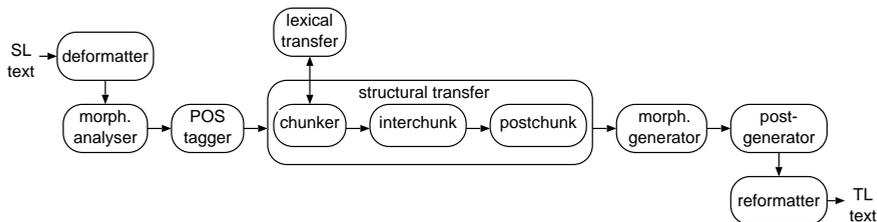
Figure 1: The modular architecture of the Apertium MT platform.

2. The Apertium platform

A very brief outline of Apertium will be given here. Turn to existing descriptions, such as Armentano-Oller et al. (2006) and Forcada et al. (2007), 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, and (d) extensive documentation on usage.[1] The Apertium engine is a pipeline or assembly line consisting of the following stages or modules (see figure 1):

- A *deformatter* which encapsulates the format information in the input document as *superblanks* that will then be seen as blanks between words by the rest of the modules.
- 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 finite-state transducer (FST) generated from a source-language (SL) morphological dictionary (MD) in XML.
- An optional *constraint grammar*[2] (Karlsson et al., 1995) to reduce or remove entirely part-of-speech (PoS) ambiguity before the statistical PoS tagger, and to provide syntactic and semantic labelling.
- A *statistical PoS tagger* which chooses, using a first-order hidden Markov model (HMM: Cutting et al. (1992)), the most likely lexical form corresponding to an ambiguous surface form, as trained using a corpus and a tagger definition file in XML.

---

[1]Documentation on a wide variety of development and usage scenarios can be found on the Apertium Wiki (http://wiki.apertium.org/).

[2]http://beta.visl.sdu.dk/constraint_grammar.html

- 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 the first module and some others call more than three, see below):
  - 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. More than one *interchunk* module can be used in sequence.
  - A *postchunk* module which performs finishing operations on each chunk and removes chunk encapsulations so that a plain sequence of lexical forms is generated.

  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 MD in XML.
- A *post-generator* which performs orthographic operations such as contractions (e.g. Spanish *del = de + el*) and apostrophations (e.g. Catalan *l'institut = el + institut*), using a FST generated from a rule file written in XML.
- A *reformatter* which de-encapsulates any format information.

3. Resources

As mentioned in the previous section, creating a machine translation system in the Apertium platform requires creating or adapting linguistic resources. As a consequence, for each of the 21 language pairs available there is at least: a finite-state morphology for analysis, another one for generation, a trained HMM-based part-of-speech tagger, a bilingual transfer lexicon,[3] and a set of transfer rules.

We describe below the current status of these resources for the platform as a whole, focussing on those resources which are stable (tested and proven). Apertium includes, in the words of Streiter, Scannell, and Stuflesser (2007), a *pool* of free resources for natural language processing targeted specifically at machine translation.

3.1. Format filters

Format filters can be used also by other MT applications. The encapsulation of formatting is simple and eases the processing of multiple document formats in an

---

[3]A bilingual transfer lexicon contains correspondences between lemmas, parts-of-speech and in some cases between other morphological features.

efficient manner. The format filters available in Apertium include ODT, HTML, RTF, MediaWiki and others. Format descriptions are based on a simple XML specification.

3.2. Morphological dictionaries

The morphological transducers used in Apertium are built using the *lttoolbox* finite-state toolkit (Ortiz-Rojas, Forcada, and Ramírez-Sánchez, 2005). The toolkit provides: a compiler, to transform the dictionaries described in XML into the fast, compact finite-state transducers that are then used by the engine.

Morphological dictionaries (MDs) are written in a format (see Forcada et al. (2007) for details) that allows users to encode regularities in the form of paradigms that may in turn call other paradigms. The compiler takes advantage of this and builds the finite-state transducer recursively, performing local minimization at each step (Ortiz-Rojas, Forcada, and Ramírez-Sánchez, 2005).

It is worth noting during the discussion of MDs that there are many languages covered where the morphology in Apertium does not provide the widest coverage for a given language. This is certainly the case for English and Spanish. However, they are included as the uniform nature of the formats and tagsets can facilitate performing experiments, and the single licence (the GNU General Public Licence[4] (GPL) used throughout) ease their integration with other free software.

Table 1 gives a breakdown of the MDs currently available and some statistics of coverage. Some of these have been built from existing resources such as the the *Norsk Ordbank* (`http://www.edd.uio.no/prosjekt/ordbanken/`), *Eurfa* (`http://kevindonnelly.org.uk/eurfa/`), *Gramadóir* (`http://borel.slu.edu/gramadoir/`), or *Matxin* (`http://matxin.sf.net`). Numbers of lemmata are approximate and include multi-word units encoded in the lexicon, the lemmata of surface forms with attached clitics and, in some cases, duplicate entries for differing orthographies.

The surface column gives the total number of surface forms recognised by the analyser. The *mean ambig.* column gives the mean ambiguity for each surface form, that is the mean number of lexical forms (analyses) returned per surface form. This gives an indication of the completeness of the morphology, although in the case of languages with prefix inflection, such as Afrikaans and Persian, the dictionary may recognise surface forms that will never appear in running texts (overanalysis).

The coverage column gives *naïve coverage*, that is, the fraction of surface forms in a representative corpus for which at least one analysis is returned. The list of analyses returned may not be complete, hence the word *naïve*. Finally the corpus column gives details of the corpus on which the evaluation was performed, WP stands for Wikipedia and is followed by the date of the database dump,[5] EP stands for EuroParl (Koehn, 2005) and is followed by the release date. These corpora were chosen as they are

---

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

[5]`http://download.wikipedia.org/`

Language	Lemmata	Surface	Mean ambig.	Coverage	Corpus
N. Nynorsk[1] (`nn`)	47,193	402,096	1.33	89.6%	WP 2009-01-19
N. Bokmål[1] (`nb`)	46,945	571,411	1.30	88.2%	WP 2009-01-08
English (`en`)	33,033	75,761	1.23	95.2%	EP 2007-09-28
Afrikaans (`af`)	14,033	42,107	1.25	80.0%	WP 2009-07-31
Danish (`da`)	10,659	80,106	1.15	86.2%	EP 2007-09-28
Icelandic (`is`)	7029	206,353	2.41	82.0%	WP 2008-03-20
Swedish (`sv`)	5,130	37,191	1.08	80.0%	EP 2007-09-28
Asturian (`ast`)	46,550	13,549,353	1.16	86.3%	WP 2009-11-17
Spanish (`es`)	41,735	4,600,370	1.40	97.6%	EP 2007-09-28
Catalan (`ca`)	37,635	7,185,455	1.15	89.8%	WP 2009-10-10
French (`fr`)	28,691	275,007	1.32	95.6%	EP 2007-09-28
Galician (`gl`)	21,298	9,764,319	1.30	86.6%	WP 2009-02-01
Romanian (`ro`)	18,719	612,511	1.28	83.6%	WP 2009-11-23
Occitan (`oc`)	18,079	6,084,575	1.05	81.0%	WP 2009-11-23
Portuguese (`pt`)	11,156	9,330,910	1.78	94.9%	EP 2007-09-28
Italian (`it`)	10,117	462,319	1.25	88.8%	EP 2007-09-28
Breton (`br`)	13,999	278,279	1.10	87.6%	WP 2009-11-11
Welsh[2] (`cy`)	11,081	438,856	1.21	86.1%	WP 2009-11-10
Irish[3] (`ga`)	8,769	165,787	1.53	83.6%	WP 2009-11-08
Persian (`fa`)	11,087	514,539	1.06	80.0%	WP 2007-11-02
Bulgarian (`bg`)	14,413	169,121	1.11	80.5%	WP 2008-05-25

1. From *Norsk Ordbank* 2. From *Eurfa* 3. From *An Gramadóir* 4. From *Matxin*

Table 1: Statistics on Apertium finite-state morphological dictionaries organised by language family

available under free licences and are widely used in machine translation.

3.3. Bilingual lexica

Along with morphological analysers, Apertium also has a number of bilingual lexica. These are encoded in the same XML-based format used by the morphological analysers, but represent correspondences between lemmata, including multi-word units, parts of speech and, in some cases, morphological information (e.g. to specify changes in the inflection information from SL to TL, and also to mark some ambiguities that should be solved by the structural transfer module).

A summary of the available bilingual lexica in Apertium can be found in table 2. Included are dictionaries which are either in released language pairs, or otherwise considered reasonably stable.

Pair	Entries	Pair	Entries	Pair	Entries	Pair	Entries
fr--ca	10,554	es--ca	40,446	en--gl	31,286	en--es	27,540
en--ca	24,601	fr--es	23,295	es--ro	21,511	oc--ca	18,896
es--it	17,294	oc--es	15,772	br--fr	15,762	es--ast	13,778
eu--es	12,174	pt--gl	11,844	es--pt	11,447	cy--en	11,405
sv--da	11,398	es--gl	10,807	pt--ca	7,716	is--en	5,875
nn--nb	73,809	ga--gd	7,863				

Table 2: Statistics on bilingual lexica available in Apertium as of November 11, 2009 (ISO-639 codes in Table 1; `ga`: Irish, `gd`: Scottish Gaelic)

3.4. Part-of-speech taggers

Apertium uses a first-order (bigram) HMM-based POS tagger (Cutting et al., 1992)[6] that is trained from corpora and a tagger definition file (see below). It can be trained using classical methods ---either supervised or unsupervised (Baum-Welch algorithm)--- or by means of a novel unsupervised approach that uses the rest of the MT engine and a TL model to estimate the HMM parameters (Sánchez-Martínez, Pérez-Ortiz, and Forcada, 2008).[7] An XML-based tagger definition file is used to specify how the lexical forms delivered by the morphological analyser must be grouped into coarse tags. Grouping lexical forms (consisting of a lemma and morphological information making up a rather ``fine-grained'' PoS tag) into coarse PoS tags is needed to reduce the amount of parameters of the HMM. Each coarse tag is defined by means of a list of fine-grained tags in which wild-cards can be used. Lexicalised coarse tags (Pla and Molina, 2004) may be defined where needed by specifying the lemma of the word in the corresponding attribute. HMM observable outputs are all the possible *ambiguity classes*, or sets of coarse tags occurring in the dictionary, plus a reasonable *open set* for unknown words.

It is also possible to define constraint rules in the form of *forbid* and *enforce* rules. *Forbid* rules define restrictions as sequences of two coarse tags that cannot occur. *Enforce* rules are used to specify the set of coarse tags allowed to occur after a particular coarse tag. These rules are applied to the HMM parameters by introducing quasi-zeroes in the state transition probabilities of forbidden sequences and re-normalising.

3.5. Transfer rules

Transfer rules for each of the three transfer stages, *chunker*, *interchunk*, and *postchunk* are written using a very similar syntax. The rules are based on finite-state pattern

---

[6]A second-order (trigram) tagger is currently under development.

[7]A free/open-source implementation is provided by package `apertium-tagger-training-tools`.

matching and are non-recursive. They are largely hand-written (but see 4.2). *Chunker* rules deal with local phenomena such as number and gender agreement in noun phrases, local word reorderings, some lexical changes (e.g. of prepositions). *Interchunk* rules are used for analogous longer-range phenomena (such as the reordering of complete chunks) and can also be used to merge chunks; *Postchunk* may be used for internal adjustments after application of *interchunk* rules.

The average number of rules per direction in a language pair with multi-stage transfer is approximately 300, and in single stage transfer around 100. For example, the Spanish to Catalan direction has 104 single-stage rules, where the English to Catalan direction has 227 *chunker* rules, 59 *interchunk* rules and 38 *postchunk* rules.

4. Reuse and recycle

This section gives a review of ways in which the resources available in Apertium can be re-used in other MT systems, for example those based on the Moses (Koehn et al., 2007) statistical MT system, and how other machine translation systems can be used to create or improve resources for Apertium.

4.1. Reuse of resources in other systems

As described in Tyers (2009), the dictionaries of Apertium (sections 3.2 and 3.3), together with very basic transfer rules can be used to create full-form bilingual vocabulary lists which can be added to an existing parallel corpus for training a statistical machine translation system based on Moses. The idea of this list is to improve coverage of word forms for inflected languages, when using a small corpus, or when the corpus is of a limited domain (for example generating second-person singular forms of verbs where the corpus contains overwhelmingly third-person singular).

Adding the dictionary also eases the computation of accurate word alignments since one-to-one word mappings are explicitly provided. In Sánchez-Martínez and Forcada (2009), table 4 (p. 22) results are given for an SMT system trained on a small corpus when the generated bilingual corpus is added and when it is not.

4.2. Corpus-based creation and improvement of resources

A corpus-based approach to infer shallow structural transfer rules is proposed by Sánchez-Martínez and Forcada (2009).[8] The authors extend the alignment template approach (Och and Ney, 2004) used in statistical MT with a set of restrictions derived from the bilingual dictionary of Apertium to control their application as transfer rules. For the translation between closely-related languages, the authors report an improvement over word-for-word translation and a translation quality close to the one

---

[8]A free implementation is provided by package `apertium-transfer-tools`.

provided by hand-coded transfer rules. Their approach also provides better translation results than the Moses statistical MT system trained on the same small parallel corpus when it is extended with the Apertium bilingual dictionary (see Section 4.1).

It is worth noting that there has been another approach to the inference of shallow structural transfer rules using corpora and Apertium resources (Caseli, Nunes, and Forcada, 2006) which, in addition to transfer rules, also automatically infers Apertium bilingual lexica.[9]

4.3. Hybridisation of Apertium and other machine translation systems

Sánchez-Martínez, Forcada, and Way (2009) have tested the integration of sub-sentential translation units (bilingual chunks) into the Apertium MT engine.[10] In their approach the bilingual chunks were automatically obtained from parallel corpora by using the marker-based chunkers and sub-sentential aligners used in the example-based MT system MaTrEx (Gough and Way, 2004, Tinsley et al., 2008).[11] Note, however, that bilingual chunks obtained in a different way could have been used, for instance the chunks[12] extraction algorithm (Zens, Och, and Ney, 2002) used by state-of-the-art statistical MT systems such as Moses.

In the integration of bilingual chunks into a rule-based MT system like Apertium, special care must be taken so as not to break the application of structural transfer rules, since this would increase the number of ungrammatical translations. Thanks to the modular design of Apertium this has been possible by developing a wrapper around the translation engine. The approach consists of (i) the application of a dynamic-programming algorithm to compute the best translation coverage of the input sentence given the collection of bilingual chunks available; (ii) the translation of the input sentence as usual by Apertium; and (iii) the application of a language model to choose one of the possible translations for each of the bilingual chunks detected. Sánchez-Martínez, Forcada, and Way (2009) report improvements, although not statistically significant, in the translation from English to Spanish, and vice versa.

5. Discussion

We have presented in this paper the resources available in the Apertium machine translation platform, and some possible uses of these resources in improving other MT systems, or creating hybrid systems. The resources we present are currently used in 21 released rule-based machine translation systems.[13] Future research is aimed

---

[9]A free/open-source implementation can be downloaded from `http://retratos.sf.net`.

[10]A free/open-source implementation is provided by package `apertium-chunks-mixer`.

[11]Selected components from MaTrEx will soon be made available as the free/open-source package *Marclator* at `http://www.computing.dcu.ie/~mforcada/marclator.html`.

[12]Usually referred as *phrases* by statistical MT practitioners.

[13]A full list may be found on the front page of the Apertium Wiki (`http://wiki.apertium.org`).

at: expanding the number of languages covered by the linguistic resources, increasing the number of language pairs, implementing a module for lexical selection, integrating other free/open-source software, such as HFST[14] or *foma* (Huldén, 2009) for managing more complex morphologies, and the implementation of a module for deeper structural transfer. Improving integration with other free/open-source machine systems such as Moses, Cunei and Matxin is also a priority.

Bibliography

Armentano-Oller, Carme, Rafael C. Carrasco, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, and Miriam A. Scalco. 2006. Open-source Portuguese--Spanish machine translation. In R. Vieira, P. Quaresma, M.d.G.V. Nunes, N.J. Mamede, C. Oliveira, and M.C. Dias, editors, *Computational Processing of the Portuguese Language, Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006*, volume 3960 of *Lecture Notes in Computer Science*. Springer-Verlag, May, pages 50--59.

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

Cutting, D., J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference*, pages 133--140, Trento, Italy, 31 mar--3 apr.

Forcada, Mikel L., Boyan Ivanov Bonev, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Carme Armentano-Oller, Marco A. Montava, and Francis M. Tyers. 2007. Documentation of the open-source shallow-transfer machine translation platform Apertium. `http://xixona.dlsi.ua.es/~fran/apertium2-documentation.pdf`, May.

Forcada, Mikel L., Francis M. Tyers, and Gema Ramírez-Sánchez. 2009. The free/open-source machine translation platform Apertium: Five years on. In F.M. Tyers J.A. Pérez-Ortiz, F. Sánchez-Martínez, editor, *Proceedings of theFirst International Workshop on Free/Open-Source Rule-Based Machine Translation FreeRBMT'09*, pages 3--10, November.

Gough, N. and A. Way. 2004. Robust large-scale EBMT with marker-based segmentation. In *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation (TMI-04)*, pages 95--104, Baltimore, MD.

---

[14]`http://www.ling.helsinki.fi/kieliteknologia/tutkimus/hfst/`

Huldén, Måns. 2009. Foma: a finite-state compiler and library. *EACL 2009*, pages 29--32.

Karlsson, F., A. Voutilainen, J. Heikkilä, and A. Anttila. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Natural Language Processing, No 4*. Mouton de Gruyter, Berlin and New York.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. *MT Summit 2005*.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007*.

Och, F. J. and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417--449.

Ortiz-Rojas, Sergio, Mikel L. Forcada, and Gema Ramírez-Sánchez. 2005. Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas. *Procesamiento del Lenguaje Natural*, (35):51--57.

Pla, F. and A. Molina. 2004. Improving part-of-speech tagging using lexicalized HMMs. *Journal of Natural Language Engineering*, 10(2):167--189, June.

Sánchez-Martínez, Felipe and Mikel L. Forcada. 2009. Inferring shallow-transfer machine translation rules from small parallel corpora. *Journal of Artificial Intelligence Research*, 34:605--635.

Sánchez-Martínez, Felipe, Mikel L. Forcada, and Andy Way. 2009. Hybrid rule-based -- example-based MT: Feeding apertium with sub-sentential translation units. In Mikel L. Forcada and Andy Way, editors, *Proceedings of the 3rd Workshop on Example-Based Machine Translation*, pages 11--18, Dublin, Ireland, November.

Sánchez-Martínez, Felipe, Juan Antonio Pérez-Ortiz, and Mikel L. Forcada. 2008. Using target-language information to train part-of-speech taggers for machine translation. *Machine Translation*, 22(1-2):29--66.

Streiter, Oliver, Kevin P. Scannell, and Mathias Stuflesser. 2007. Implementing NLP Projects for Non-Central Languages: Instructions for Funding Bodies, Strategies for Developers. *Machine Translation*, 20(4):267--289.

Tinsley, J., Y. Ma, S. Ozdowska, and A. Way. 2008. Matrex: the DCU MT system for WMT 2008. In *Proceedings of the Third Workshop on Statistical Machine Translation, ACL 2008*, pages 171--174, Columbus, OH.

Tyers, Francis M. 2009. Rule-based augmentation of training data in Breton–French statistical machine translation. In *Proceedings of the 13th Annual Conference of the European Association of Machine Translation, EAMT09*, pages 213--218.

Zens, R., F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *KI 2002: Advances in Artificial Intelligence: Proceedings 25th Annual German Conference on AI*, volume 2479 of *Lecture Notes in Computer Science*. Springer-Verlag, pages 18--32.