Machine Translation manuscript No.
(will be inserted by the editor)

Apertium: a free/open-source platform for rule-based machine translation

Mikel L. Forcada · Mireia Ginestí-Rosell · Jacob Nordfalk · Jim O'Regan · Sergio Ortiz-Rojas · Juan Antonio Pérez-Ortiz · Felipe Sánchez-Martínez · Gema Ramírez-Sánchez · Francis M. Tyers

Version: Thursday 3rd March, 2011, 14:28

Abstract Apertium is a free/open-source platform for rule-based machine translation. It is being widely used to build machine translation systems for a variety of language pairs, especially in those cases (mainly with related-language pairs) where shallow transfer suffices to produce good quality translations, although it has also proven useful in assimilation scenarios with more distant pairs involved. This paper summarises the Apertium platform: the translation engine, the encoding of linguistic data, and the tools developed around the platform. The present limitations of the platform and the challenges posed for the coming years are also discussed. Finally, evaluation results for some of the most active language pairs are presented. An appendix describes Apertium as a free/open-source project.

Keywords Free/open-source machine translation · Rule-based machine translation · Apertium · Shallow transfer · Finite-state transducers

1 Introduction

We briefly describe Apertium, a free/open-source (FOS) machine translation (MT) platform comprising an engine, a toolbox, and data to build rule-based MT systems. The platform was initially aimed at related-language pairs (such as Spanish–Portuguese) but it was expanded later to deal with more divergent pairs (such as English–Catalan). Apertium uses finite-state transducers (Roche and Schabes, 1997) for lexical processing, hidden Markov models for part-of-speech tagging (Cutting et al., 1992), and multi-stage finite-state *chunking* for structural transfer. It may be used to

M.L. Forcada, M. Ginestí, J.A. Pérez-Ortiz, F. Sánchez-Martínez, F.M. Tyers
Grup Transducens, Dept. Llenguatges i Sistemes Informátics, Universitat d'Alacant, Spain

S. Ortiz-Rojas, G. Ramírez-Sánchez
Prompsit Language Engineering, Elx, Spain

J. Nordfalk
Copenhagen University College of Engineering, Denmark

J. O'Regan
Eolaistriu Technologies, Thurles, Ireland

Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J. et al. Apertium: a free/open-source platform for rule-based machine translation. Machine Translation 25, 127–144 (2011).

build MT systems for a variety of language pairs; to that end, the platform uses simple, standard formats to encode the linguistic data needed, and documented procedures to build those data and to train the necessary modules. Apertium is licensed under the GNU General Public License[1] (GNU GPL) and can be downloaded from the project's website: `http://www.apertium.org`.

MT software is special in the way it strongly depends on data. On the one hand, rule-based MT (RBMT) depends on explicit linguistic data such as morphological dictionaries, bilingual dictionaries, grammars, and structural transfer rules (see Hutchins and Somers, 1992, ch. 4); on the other hand, corpus-based MT depends, directly or indirectly, on the availability of sentence-aligned *parallel* (bilingual) texts. Development and research on corpus-based MT, mainly statistical machine translation (SMT; Koehn, 2010), has drastically increased since the nineties, due to the significant rise in computational power and storage capacity of modern computers, and the growing availability of parallel texts. Both SMT and RBMT paradigms have pros and cons and neither of them can be identified as inherently better than the other; in fact, hybridisation is currently an active field of research (Thurmair, 2009). As a FOS RBMT system, Apertium offers some competitive advantages over SMT:

- SMT systems often output translations which are *more natural* than those produced by RBMT systems, but less faithful to the original. SMT attempts to balance, on one hand, the probability that the words of the translation correspond to those of the original sentence (*fidelity*) and, on the other hand, the probability that the words of the translated sentences are those and in that order in the target language (*fluency*). It happens sometimes that the latter outweighs the former: the result is a deceptively fluent translation which, however, is not faithful to the original.
- RBMT systems tend to produce translations which are more *mechanical*, sometimes less fluid and more *repetitive*, so that their errors tend also to be more repetitive (Guzmán, 2008) and usually very evident, due the absence of any mechanism for *smoothing* the resulting translation to make it more *fluent*. This eases the work of posteditors, who tend to prefer MT systems that are predictable (Koehn, 2010, p. 222) because of being repetitive (Way, 2010, Sect. 3.4).
- Another advantage of the RBMT systems is *terminological consistency*. Whereas RBMT systems produce the same equivalent (or an equivalent from a small list of candidates if the system includes a module for lexical selection) for the same words across the text, SMT systems may translate the same word in different seemingly random ways as they choose translation equivalents according to the translation probability of the whole sentence, or may have been trained on corpora which are not entirely parallel.[2]
- Experts who have designed a system based on rules find it much easier to diagnose and repair the source of a translation error: they may easily discover which rule has failed (specially, when the number of rules in the system is not very large) or which entry in the dictionary is wrong.
- When building RBMT systems, linguistic knowledge for a language pair is encoded explicitly in the form of linguistic data. This makes them naturally available to build knowledge for other language pairs or even for other human language technologies

[1] `http://www.gnu.org/licenses/#GPL`

[2] See `http://itre.cis.upenn.edu/~myl/languagelog/archives/005494.html` for an example of this behaviour.

(especially when FOS licences are involved), and, conversely, linguistic knowledge from other sources may be reused to build MT systems.
– Finally, Apertium eases the development of MT systems for the translation between less-resourced languages, and also between morphologically rich languages, which, in a corpus-based MT setting, even with large corpora, may suffer from data sparseness. It is quite hard to obtain and prepare the amounts of sentence-aligned parallel text (of the order of hundreds of thousands or millions of words) required to get reasonable results in *pure* corpus-based MT; however, it may be much easier for speakers to encode the language expertise needed to build an RBMT system.

Apertium is not the only MT system that has been released under an FOS license. Among the RBMT systems we find:

– The FOS version of the commercial MT system Logos, known as OpenLogos (Scott and Barreiro, 2009). There are data available for English and German as source languages and French, Spanish, Italian and Portuguese as target languages.
– Anusaaraka (Chaudhury et al., 2010) has evolved from a script translator into a full FOS RBMT system from English into Hindi.
– Matxin (Alegria et al., 2007) uses deeper syntactic representations than Apertium and has been developed with the Spanish–Basque pair in mind, although it could, in principle, be adapted to different language pairs (Mayor and Tyers, 2009); some of the components of Matxin come from the Apertium platform.
– Bond et al. (2005) showed how they could build a partially FOS Japanese–English system combining different available technologies and linguistic resources.
– Other attempts at FOS implementations of MT systems were initiated, such as GPLTrans[3] (project idle since 2002), Traduki[4] (project idle since 2004) and Linguaphile[5] (project almost idle).

There are also many FOS corpus-based MT systems such as the phrase-based and tree-based SMT system Moses (Koehn et al., 2007), the hybrid example-based–SMT system Cunei (Phillips, 2007), and the tree-based SMT system Joshua (Li et al., 2009).[6]

This article compiles some of the previous work published about Apertium and integrates it into a general, up-to-date overview of the platform. The paper is organised as follows. Sec. 2 describes the Apertium translation engine. After that, Sec. 3 introduces the existing linguistic resources for Apertium and how they are encoded. Then, Sec. 4 introduces the compilers that convert linguistic data into an efficient binary format used by the engine, and other tools that ease the development of new data. After that, evaluation results for some of the most active language pairs are reported in Sec. 5. The paper ends with some concluding remarks and an appendix that briefly describes the objectives and community of the Apertium project.

2 Apertium engine

The MT engine and tools in Apertium were not built from scratch, but are rather the result of a complete rewriting and extension of two previous MT systems, namely

[3] `http://www.translator.cx/`
[4] `http://traduki.sourceforge.net/`
[5] `http://linguaphile.sourceforge.net/`
[6] See `http://fosmt.org` for a more complete list of FOS MT systems.

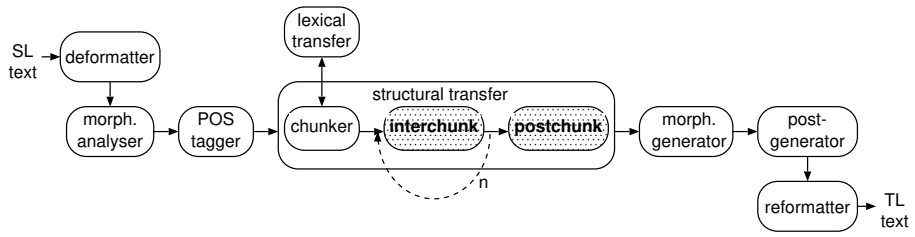

Fig. 1 The Apertium architecture. Shadowed modules are optional and intended for less-related pairs. Apertium level 2 allows for an arbitrary number of interchunk modules.

the Spanish–Catalan MT system `interNOSTRUM.com` (Canals-Marote et al., 2001) and the Spanish–Portuguese MT system `traductor.universia.net` (Garrido-Alenda et al., 2004), both developed by the Transducens group at Universitat d'Alacant. The first version of the whole system (Apertium level 1) was released on July 29, 2005, and closely followed the architecture of those two non-free systems. An enhanced version of the engine (Apertium level 2) was released on December 22, 2006, featuring an extended implementation of the structural transfer of Apertium level 1 to perform more complex transformations for the translation between less-related language pairs.

2.1 Translation pipeline

The Apertium translation engine consists of a Unix-style *pipeline* or *assembly line* with the following modules (see Fig. 1):

- A *deformatter* which encapsulates the format information in the input 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 (SF) (*words*, or, where detected, multi-word lexical units or MWLUs) and delivers, for each of them, one or more *lexical forms* (LF) consisting of *lemma*, *lexical category* and morphological information. It reads a finite-state transducer (FST) compiled from a source-language (SL) morphological dictionary in XML.
- A *statistical PoS tagger* which chooses, using a first-order hidden Markov model (HMM) (Cutting et al., 1992), the most likely LF corresponding to an ambiguous SF.
- A *lexical transfer* module which reads each SL LF and delivers the corresponding target-language (TL) LF by looking it up in a bilingual dictionary encoded as an FST compiled from the corresponding XML file.
- A *structural transfer* module which consists of three sub-modules:
  - A mandatory *chunker* which, after invoking the 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 optional *interchunk* module which performs longer-range operations with the chunks and between them. More than one *interchunk* module can be used in sequence to perform increasingly higher-level transfer transformations.

Table 1 An example of step-by-step execution of Apertium when translating the HTML text "We will go to the old park" into Spanish. The output of each module becomes the input of the next one (see text for details).

Module producing output	Output
Deformatter	`We will go to the[ <b>]old park[</b>]`
Morphological analyser	`^We/Prpers<prn><subj><p1><mf><pl>$`
	`^will/will<n><sg>/will<vaux><inf>$`
	`^go/go<vblex><inf>/go<vblex><pres>$ ^to/to<pr>$`
	`^the/the<det><def><sp>$[ <b>]^old/old<adj><sint>$`
	`^park/park<n><sg>/park<vblex><inf>/park<vblex><pres>$[</b>]`
Part-of-speech tagger	`^Prpers<prn><subj><p1><mf><pl>$ ^will<vaux><inf>$`
	`^go<vblex><inf>$ ^to<pr>$ ^the<det><def><sp>$`
	`[ <b>]^old<adj><sint>$ ^park<n><sg>$[</b>]`
Chunker (transfer)	`^Prnsubj<SN><tn><p1><GD><pl>{^prpers<prn><2><p1><4><pl>$}$`
	`^verbcj<SV><vblex><fti><PD><ND>{^ir<vblex><3><4><5>$}$`
	`^pr<PREP>{^a<pr>$}$`
	`^det_nom_adj<SN><DET><m><sg>{^el<det><def><3><4>$`
	`[ <b>]^parque<n><3><4>$ ^viejo<adj><3><4>$}$[</b>]`
Interchunk (transfer)	`^Verbcj<SV><vblex><fti><p1><pl>{^ir<vblex><3><4><5>$}$`
	`^pr<PREP>{^a<pr>$}$`
	`^det_nom_adj<SN><DET><m><sg>{^el<det><def><3><4>$`
	`[ <b>]^parque<n><3><4>$ ^viejo<adj><3><4>$}$[</b>]`
Postchunk (transfer)	`^Ir<vblex><fti><p1><pl>$ ^a<pr>$ ^el<det><def><m><sg>$`
	`[ <b>]^parque<n><m><sg>$ ^viejo<adj><m><sg>$[</b>]`
Morphological generator	`Iremos ~a el[ <b>]parque viejo[</b>]`
Postgenerator	`Iremos al[ <b>]parque viejo[</b>]`
Reformatter	`Iremos al <b>parque viejo</b>`

  - An optional *postchunk* module which performs finishing operations on each chunk and removes chunk encapsulations so that a plain sequence of LFs is generated.

  Some language pairs use only the first one (*chunker*), which is equivalent to Apertium level 1, while others use one or more *interchunk* submodules and an additional *postchunk* submodule (Apertium level 2). All of these modules are compiled from files containing rules that associate an *action* to each defined LF *pattern*. Patterns are applied left-to-right, and the longest matching pattern is always selected.
- A *morphological generator* which delivers a TL SF for each TL LF, by suitably inflecting it. It reads an FST compiled from a TL morphological dictionary in XML.
- A *post-generator* which performs orthographic operations, such as contractions (e.g. Spanish *de + el = del* or Portuguese *por + as = pelas*), apostrophations (e.g. Catalan *el + institut = l'institut*) or epenthesis (e.g. English *a + institute = an institute*), using an FST generated from a rule file written in XML.
- A *reformatter* which de-encapsulates any format information.

2.2 Translation example

Table 1 shows the output of each module in the Apertium pipeline (see Fig. 1) when translating one sentence written in HTML from English to Spanish. First, the deformatter encapsulates format information (in this case, HTML tags) in square brackets,

so that the rest of the modules treat it as simple blanks between words. Then, the morphological analyser delivers one LF for each of the unambiguous input SFs, and two or more for the SFs that, according to the English monolingual dictionary, may be assigned different lexical categories (*will* can be a noun or an auxiliary verb; *go* can be a verb in infinitive or in present tense; *park* can be a noun or a verb, in infinitive or in present tense); the rest of the words are tagged as subject pronoun (*we*), preposition (*to*), definite determiner singular/plural (*the*), and synthetic adjective (*old*).[7] The characters "`^`" and "`$`" delimit the analysis for each SF, and the different LFs for each SF are separated by "`/`". The string after the "`^`" and before the first "`/`" is the SF as it appears in the input text; the string before each group of lexical labels is the lemma. In the next step, the ambiguous words are correctly tagged by the part-of-speech tagger.

The chunker detects patterns of words, creating four chunks in this case; it also calls the lexical transfer module to obtain the corresponding LFs in Spanish for each English LF. The chunker executes the local actions programmed for each detected pattern, which can imply local reorderings, deletion or insertion of words. Here, the chunk labelled `verbcj` is generated for the detected sequence *auxiliary verb–verb* (*will go*), and it contains only one LF, the Spanish verb *ir*; the auxiliary is used to determine the value *fti* (future) of the verb chunk. The sequence *determiner–adjective–noun* (*the old park*) is labelled `det_nom_adj`, and the adjective is moved after the noun. Two other chunks are generated, one for the pronoun (labelled `Prnsubj`) and another for the preposition (labelled `pr`). The LFs belonging to each chunk are enclosed between curly brackets, and the labels outside correspond to the lexical information from the head of the TL chunks (for example, the noun in the noun phrase) or, in the absence of this information, from some of the other constituents in order of *importance*. Note that the labels with numbers link the grammatical information of elements inside the chunk to that of elements outside the chunk. This is how the postchunk module will be able to determine later that *el* and *viejo* must be assigned the tags `m` (masculine) and `sg` (singular) to match the gender and number of the noun *parque*, or that the verb *ir* must be assigned the future tense (`fti`). Note also that the labels `GD`, `PD` and `ND` in the first and second chunks (meaning *gender to be determined, person to be determined* and *number to be determined*) indicate that there was not enough information at chunk level to determine this grammatical information, so that the task is passed on to the next module, where operations between chunks can be performed.

The interchunk module detects the sequence `Prnsubj`–`verbcj` and uses the grammatical information of the pronoun chunk to assign person and number to the verb chunk, so that `PD` is now *first person* (`p1`) and `ND` is now *plural* (`pl`). It also deletes the pronoun chunk.

In the generation phase, the morphological generator delivers a TL SF for each TL LF by looking them up in the Spanish monolingual dictionary. After that, the postgenerator performs the contraction of *a+el* into *al*. Finally, the reformatter restores the format information (HTML tags) into the translated text.

2.3 Limitations and work ahead

The Apertium engine 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:

[7] *Synthetic* adjectives, such as *old*, are inflected for comparison by adding a morpheme (i.e. *old*, *older*, *oldest*) in opposition to analytic adjectives, e.g. *expensive*, that are not inflected.

- The performance of the part-of-speech tagging module is below the state of the art for many languages. However, recently an optional *constraint grammar*[8] module (Karlsson, 1995) has been integrated before the tagger to reduce or entirely remove part-of-speech ambiguity.
- *Polysemous* SL words may have more than one TL equivalent. Apertium bilingual dictionaries currently provide only one TL LF per SL LF. Fixed-length MWLUs may be used to choose a different equivalent in a fixed context, but there are many cases where this is not sufficient. No successful, efficient, general-purpose lexical selection module has been implemented yet, although the dictionary format already allows more than one TL equivalent per SL lemma.
- The structural transfer component does not rely on a full parse tree of the whole sentence, but rather on one or more levels of *chunking*. Even if it is possible for a processed pattern to leave information for later patterns, which can be used for long-range agreement processes, it is still hard to deal with long-range phenomena.
- The structural transfer module is by far the most time-consuming one: it consumes around 95% of the CPU time needed to perform a translation because the XML code of transfer rules is interpreted at run-time instead of compiled into an optimised binary form. Preliminary experiments show that by translating the rules into Java code which is then compiled into bytecode and executed on the Java Virtual Machine a speedup factor of around 3 in the overall translation time is possible.[9]
- Complex and discontiguous MWLUs are not well covered by the system. There is support for MWLUs where only one part of the unit inflects (for example, contiguous phrasal verbs as *take away* in English), but discontiguous usage (*takes the rubbish out*) is not currently treatable. Contiguous MWLUs where multiple parts inflect to agree (*dirección general*, *direcciones generales* in Spanish), are supported only in a very rudimentary way (straightforward enumeration of all forms).

3 Apertium data

The initial funding for Apertium came from the Spanish Ministry of Industry, Tourism and Commerce in July 2004, which funded a consortium to develop translation technology for the languages of Spain;[10] thus, the first language pairs implemented were Spanish–Catalan and Spanish–Galician, which were built by combining in-house resources with free data from Freeling (Carreras et al., 2004). Since then, several language pairs have received funding both from public institutions such as the Generalitat de Catalunya (development of Apertium level 2 and data for Catalan–English translation) and private ones such as the Google Summer of Code programme[11] (Norwegian Bokmål–Nynorsk, Swedish–Danish, among others), just to name a few.

Table 2 Statistics on Apertium finite-state morphological dictionaries, organised by language family. This table shows statistics for released linguistic packages; preliminary resources also exist for additional languages such as Persian, Italian, Irish, Afrikaans and Bengali.

Language	Code	Lemmata	Surface	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
Danish	da	10,659	80,106	1.15	86.2%	EP 2007-09-28
Icelandic	is	9,134	279,164	2.45	83.7%	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	17,078	466,801	1.10	89.1%	WP 2009-11-11
Welsh[2]	cy	11,081	438,856	1.21	86.1%	WP 2009-11-10
Macedonian	mk	8,094	157,654	1.14	92.1%	ST -
Bulgarian	bg	7,873	142,063	1.18	88.1%	ST -
Basque[3]	eu	11,463	4,238,126	1.37	79.6%	WP 2009-04-05
Esperanto	eo	31,205	397,259	1.47	88.0%	WP 2010-01-12

1. From *Norsk Ordbank* 2. From *Eurfa* 3. From *Matxin* (see text)

3.1 Linguistic data

Table 2 enumerates the monolingual dictionaries available and some statistics of coverage. Some dictionaries have been built from existing resources such as Norsk Ordbank,[12] Eurfa,[13] or Matxin.[14] Numbers of lemmata are approximate and include MWLUs encoded in the lexicon and duplicate entries for differing orthographies.

The *surface* column gives the total number of SFs recognised by the analyser, including forms with attached clitics. The ambiguity column (*ambig.*) gives the average ambiguity for each SF, i.e. the average number of LFs (analyses) returned per SF. This gives an indication of the completeness of the morphology.

The coverage column gives *naïve coverage* (the list of analyses returned may not be complete), that is, the fraction of SF in a representative corpus for which at least one analysis is returned. Finally, the corpus column gives details of the corpus on which the statistics were calculated: WP stands for Wikipedia and is followed by the date of

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

[9] http://wiki.apertium.org/wiki/Bytecode_for_transfer/Evaluation

[10] Four languages share official status with Spanish in some areas of Spain: Basque, Galician, Catalan (also called Valencian), and Occitan (Aranese). Other languages such as Asturian or Aragonese have a more limited legal status.

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

[12] http://www.edd.uio.no/prosjekt/ordbanken/

[13] http://kevindonnelly.org.uk/eurfa/

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

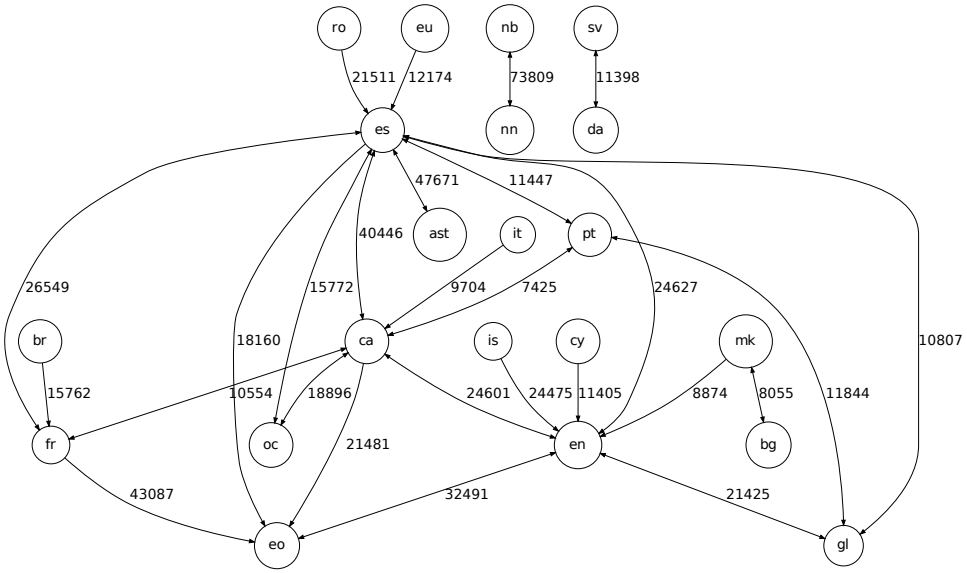

Fig. 2 The bilingual resources available in released language pairs. Edges are labelled with the number of entries contained as of 14th February, 2011 in the Apertium bilingual dictionary of the corresponding language pair (see table 2 for ISO-639 language codes).

the database dump,[15] EP stands for EuroParl (Koehn, 2005) and is followed by the release date, and ST stands for SETimes (Tyers and Alperen, 2010). These corpora were chosen as they are available under free licences and are widely used in MT.

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 MWLUs. Each bilingual correspondence is an entry in the dictionary, where lemma and part of speech are specified 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 graphical summary of the available bilingual lexica in Apertium can be found in Figure 2. We can get an idea of the most demanded language pairs by looking at the statistics of the translation requests received by the webform on the Apertium website[16] during a week. Table 3 shows these figures.

Several papers describe the creation of data for new Apertium language pairs, using a variety of approaches, including the reuse of existing FOS resources (Armentano-Oller and Forcada, 2008; Ginestí-Rosell et al., 2009; Tyers et al., 2009; Tyers and Donnelly, 2009).

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

[16] `http://www.apertium.org/`

Table 3 Distribution over language pairs of the translation requests received by the Apertium website in a week (22–29 June 2010; see table 2 for ISO-639 country codes). The *Other* group contains 31 additional language directions, including English–Spanish (1.14% of requests), Basque–Spanish (0.76%), and Catalan–French (0.14%). The total amount of the requests in this week was 27 637.

Direction	Requests	Percentage	Direction	Requests	Percentage
nn–nb	9,623	34,82%	es–pt	1,054	3,81%
es–ca	4,188	15,15%	en–eo	824	2,98%
pt–es	3,466	12,54%	gl–es	499	1,80%
es–pt_BR	1,966	7,11%	eo–en	427	1,54%
es–en	1,549	5,60%	Other	413	14,65%

3.2 Example

A small example follows to show how a simple entry is encoded in a XML monolingual dictionary. These dictionaries have basically two types of data: *paradigms*, that group regularities in inflection, and *word entries*. Once the most frequent paradigms in a dictionary are defined, entering a new inflected word is generally limited to writing the lemma and choosing an inflection paradigm. A paradigm named `par123` to be used in English nouns with singular ending in *-um* which change it to *-a* to form the plural form will be defined as follows:

```
<pardef n="par123">
 <e><p>  <l>um</l>  <r>um<s n="n"/><s n="sg"/></r>  </p></e>
 <e><p>  <l>a</l>   <r>um<s n="n"/><s n="pl"/></r>  </p></e>
</pardef>
```

Now, the words *baterium*/*bacteria* and *datum*/*data* will be defined as follows:

```
<e lm="bacterium"><i>bacteri</i><par n="par123"/></e>
<e lm="datum"><i>dat</i><par n="par123"/></e>
```

The part inside the *i* element contains the prefix of the word that is common to all inflected forms, and the element *par* refers to the inflection paradigm of the word. In this case, *bacterium* will be analysed into `bacterium<n><sg>` and *bacteria* into `bacterium<n><pl>`.

It is also possible to create entries consisting of two or more words if these words are considered to build a single *translation unit* (see MLWUs in 2.3). Dictionaries may also contain *nested paradigms* used in other paradigms (for instance, paradigms for enclitic pronoun combinations are included in all Spanish verb paradigms).

3.3 Limitations and work ahead

The current way of representing and processing lexical data in Apertium also gives rise to a number of limitations, such as the following:

– The current design of morphological analysis and generation make it hard to write morphological dictionaries for agglutinative languages such as Basque or Sámi. Also, their design is too geared toward suffix or prefix morphology, which makes it hard to treat languages with non-catenative morphology, such as Arabic.

– The management of inflection paradigms is still not powerful enough to represent all relevant regularities, although the use, in some language pairs (such as English-Catalan or Occitan-Catalan), of higher-level representations (*metadictionaries*) which are then transformed to standard Apertium dictionaries before compiling simplifies the task to some extent. Besides that, some works (Larasati and Kuboň, 2010) have recently started to integrate other morphological tools, such as HFST[17] and Foma[18], into Apertium's pipeline to handle analysis and generation of languages with complex inflection.
– Many languages, such as Icelandic or German, write many compounds as single words and do so very productively. Apertium does not have a general mechanism to segment compounds into lexical units, although some progress has been made. A compounding module has been successfully applied in the N. Nynorsk–N. Bokmål language pair; preliminary experiments show an improvement in WER around 2%.

4 Compilers and other tools

4.1 Compilers

The Apertium platform contains compilers to convert the linguistic data into the corresponding efficient (binary) form used by the modules of the engine (Ortiz-Rojas et al., 2005). Two main compilers are used: one for the four lexical processing modules of the system and another one for the structural transfer modules (see Sec. 2.1).

The lexical processor compiler is very fast (it takes about a minute to compile typical dictionaries with a number of lemmas of the order of 10,000) thanks to the use of advanced transducer building strategies and to the minimisation of partial finite-state transducers (FST) (Roche and Schabes, 1997) during construction. The four lexical processing modules (morphological analyser, lexical transfer, morphological generator, post-generator) read the resulting binary files containing a compact and efficient representation of a class of FST; in particular, augmented letter transducers (Garrido-Alenda et al., 2002). Apertium's implementation of these transducers has been optimised (Ortiz-Rojas et al., 2005) so that they are able to process tens of thousands of words per second in a current desktop computer.

The current structural transfer compiler preprocessor is used for both Apertium level 1 and level 2. It reads in a structural transfer rule file and generates a file with precompiled patterns and indexed versions of the action part of the rules to be interpreted at translation time.

4.2 Other free/open-source tools in the Apertium platform

Apart from the translation engine itself and the compilers mentioned above, other tools have been developed to ease the development of data for new language pairs, or to extend the standard behaviour of Apertium. There follows is a non-exhaustive description of some of these FOS tools:

[17] `http://www.ling.helsinki.fi/kieliteknologia/tutkimus/hfst/`
[18] `http://foma.sourceforge.net/`

- The use of `apertium-dixtools` may assist in the task of building lexical dictionaries for language pair *A*–*B* when data for *A*–*C* and *C*–*B* are available (Armentano-Oller and Forcada, 2008), but manual completion of the task by an expert is still necessary.
- Package `apertium-tagger-training-tools` implements a novel approach to train the SL part-of-speech tagger in an unsupervised way by using an unrelated corpus of TL texts and the remaining modules of the MT engine (Sánchez-Martínez et al., 2008; Sánchez-Martínez, 2008). The resulting part-of-speech tagger performs better than those trained through the classical unsupervised Baum-Welch algorithm and similarly to those trained in a supervised way from hand-tagged corpora.
- To help in the development of Apertium level 1 transfer rules, package `apertium-transfer-tools` implements an alignment-template-based approach (Och and Ney, 2004) to infer structural transfer rules from a relatively small, sentence-aligned parallel corpus (Sánchez-Martínez and Forcada, 2009; Sánchez-Martínez, 2008). The inferred rules can be then edited by a linguist, who may also add new rules if necessary, or even merged with pre-existing hand-written rules.
- Package `apertium-pn-recogniser` implements a module to be integrated in the Apertium pipeline to detect proper nouns in the input and prevent them from being translated. It is mainly based on the one already included in Freeling (Carreras et al., 2004).
- Package `apertium-chunks-mixer` allows the integration of bilingual chunks (sub-sentential translation units) obtained by aligning a parallel corpus into a translator built using Apertium (Sánchez-Martínez et al., 2009).
- Massive high-demand access to online translation services requires scalable MT systems and *application programming interfaces* (API). This has encouraged the development of ScaleMT (Sánchez-Cartagena and Pérez-Ortiz, 2010a), a framework that exposes the Apertium engine as a scalable public web service.

5 Evaluation

In this section we report translation results for some language pairs developed under the Apertium platform. Table 4 describes the corpora used to perform the evaluation together with the number of sentences and words in each language. To evaluate the Spanish–English and Spanish–French language pairs we used the test set released as part of the WMT 2010 translation task;[19] for the Spanish–Catalan we used parallel sentences from Consumer Eroski Parallel Corpus (Alcázar, 2005); for the N. Bokmål–N. Nynorsk pair we used texts from a webpage of the Norwegian Government.[20]

Table 5 reports the 95% confidence interval for the *word error rate* (WER), the translation edit rate (TER; Snover et al. (2006)), and the position-independent error rate (PER) achieved by different RBMT systems and by Apertium. Confidence intervals were calculated through the bootstrap resampling (Efron and Tibshirani, 1994) method as described by Koehn (2004). The RBMT systems to which we compare the performance of Apertium are the following closed-source commercial online systems: the Spanish–Catalan *Kwik Translator* by Lucy Software,[21] the version of Systran pro-

[19] `http://www.statmt.org/wmt10/test.tgz`
[20] `http://www.norge.no`
[21] `http://www.lucysoftware.com`

Lang. pair	Test set	Sentences	Words
es-ca	Corpus Eroski	2,400	es: 55,064; ca: 54,730
es-en	WMT10	2,489	es: 58,015; en: 54,021
es-fr	WMT10	2,489	es: 58,015; fr: 59,027
nb-nn	NORGE	500	nb: 7,260; nn: 7,371

Table 4 For each evaluation set used, number of sentences and number of words in each language (see text for details, and see table 2 for ISO-639 language codes).

Direction	MT system	WER (%)	TER (%)	PER (%)
es-ca	Apertium	[14.5, 15.6]	[13.9, 14.9]	[11.9, 12.7]
	Lucy Software	[14.2, 15.2]	[13.5, 14.5]	[11.5, 12.3]
ca-es	Apertium	[15.0, 16.0]	[14.4, 15.3]	[12.5, 13.4]
	Lucy Software	[14.9, 15.9]	[14.3, 15.2]	[12.5, 13.4]
es-en	Apertium	[73.6, 75.3]	[70.4, 72.1]	[59.4, 61.1]
	Yahoo! Babel Fish	[73.0, 74.7]	[69.6, 71.2]	[58.3, 60.0]
	Promt Translator	[71.7, 73.4]	[68.3, 69.9]	[58.0, 59.6]
en-es	Apertium	[70.1, 71.5]	[66.9, 68.2]	[54.9, 56.0]
	Yahoo! Babel Fish	[68.3, 69.8]	[64.5, 66.0]	[51.2, 52.8]
	Promt Translator	[63.3, 65.0]	[59.7, 61.3]	[46.9, 48.5]
es-fr	Apertium	[66.8, 68.4]	[63.5, 65.0]	[51.1, 52.5]
	Yahoo! Babel Fish	[65.3, 67.0]	[62.1, 63.7]	[50.0, 51.5]
fr-es	Apertium	[66.9, 68.7]	[63.4, 65.0]	[52.3, 53.8]
	Yahoo! Babel Fish	[64.3, 66.0]	[60.4, 62.1]	[48.2, 49.6]
nb-nn	Apertium	[16.5, 19.0]	[16.2, 18.6]	[14.4, 16.4]
	Nyno	[12.3, 14.9]	[12.0, 14.3]	[10.7, 12.8]

Table 5 95% confidence intervals for the word error rate (WER), the translation edit rate (TER) and the position-independent error rate (PER) when translating the test set corresponding to each pair (see Table 4).

vided by Yahoo! Babel Fish,[22] PROMT Translator,[23] and Nyno,[24] the only competing MT system for Norwegian Bokmål–Norwegian Nynorsk to our knowledge.

Results in Table 5 show that, with the exception of the English–Spanish and N. Bokmål–N. Nynorsk translation tasks, Apertium achieves results similar to those achieved by the closed-source, commercial systems we have used. The large difference in performance between es-ca and nb-nn on one hand, and the rest of language pairs on the other can be put down to the distances between the languages involved and, upon manual inspection, the fact that in the first case the reference translations were more literal.

6 Concluding remarks

We have given an overview of Apertium, a FOS platform to build rule-based MT systems, which provides a MT engine, linguistic data (dictionaries and rule files), tools to manage those data and compile them to the representation used by the engine and

[22] http://babelfish.yahoo.com

[23] http://www.online-translator.com

[24] http://nyno.no

a variety of other tools. We have described in particular detail the currently available language-pair resources, and have evaluated some of those language pairs against commercial rule-based MT systems with encouraging results.

Acknowledgements We thank the support of the Spanish Ministry of Science and Innovation through project TIN2009-14009-C02-01. Apertium has been mainly 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, the Universidade de Vigo, Ofis ar Brezhoneg and Google Summer of Code (2009 and 2010 editions). Many companies have also invested in it: Prompsit Language Engineering, ABC Enciklopedioj, Eleka Ingeniaritza Linguistikoa, imaxin|software, etc. We also thank all of the independent developers that have made substantial contributions.

A Apertium as a free/open-source project

According to the by-laws of the Apertium project,[25] its mission is to collaboratively develop FOS MT for as many languages as possible, and in particular:

1. To give everyone free, unlimited access to the best possible MT technologies.
2. To maintain a modular, documented, open platform for MT and other human language processing tasks.
3. To favour the interchange and reuse of existing linguistic data.
4. To make integration with other FOS technologies easier.
5. To radically guarantee the reproducibility of MT and natural language processing research.

The fact that the linguistic skills needed for writing resources for Apertium are relatively simple and that the whole platform is FOS has contributed to the consolidation of an active community of developers and users, especially speakers of less-resourced languages, often forgotten by the mainstream commercial MT systems. A group of more than 100 developers,[26] most of them from outside the original group, has formed around the platform. A *Project Management Committee* and an *Assembly of Committers* constitute the main governing boards of the project.

Code, especially language-pair data, is updated very frequently: hundreds of monthly commits are made to the project's repository. A collectively-maintained *wiki*[27] 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 official Apertium mailing list[28] has received more than 3 600 messages from May 2007 to July 2010 (almost 100 messages per month on average). The strength of the Apertium community may also be measured by the large number of externally developed tools and code that add functionalities to the platform. The Apertium project has been assigned 9 students in both 2009 and 2010 editions of the Google Summer of Code program.[29]

Besides that, with the mission of easing the development of new language pairs through the Internet and foster large-scale collaboration between Apertium users, a web application aimed at providing a social translation platform for Apertium has recently started to be implemented (Sánchez-Cartagena and Pérez-Ortiz, 2010b).

Finally, members of the Apertium community have also packaged the stable Apertium components for Debian GNU/Linux[30] and as a result, Apertium is part of the popular Ubuntu[31] distribution.

[25] `http://wiki.apertium.org/wiki/By-laws`

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

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

[28] `https://lists.sourceforge.net/lists/listinfo/apertium-stuff`

[29] `http://code.google.com/soc/`

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

[31] `http://www.ubuntu.com/`

References

Alcázar, A. (2005). Towards linguistically searchable text. In *Proceedings of BIDE (Bilbao-Deusto) Summer School of Linguistics 2005*, Bilbao. Universidad de Deusto.

Alegria, I., de Ilarraza, A., Labaka, G., Lersundi, M., Mayor, A., and Sarasola, K. (2007). Transfer-based MT from Spanish into Basque: reusability, standardization and open source. *Lecture Notes in Computer Science*, 4394:374–384.

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

Bond, F., Oepen, S., Siegel, M., Copestake, A., and Flickinger, D. (2005). Open source MT with DELPH-IN. In *OSMaTran, A workshop at MT Summit X*, pages 15–22.

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*.

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.

Chaudhury, S., Sharma, D., and Kulkarni, A. (2010). Anusaaraka: an approach to machine translation. In *Proceedings of the International Conference on Language, Society and Culture in Asian Context*.

Cutting, D., Kupiec, J., Pedersen, J., and Sibun, P. (1992). A practical part-of-speech tagger. In *3rd Conf. on Applied NLP. Association for Comp. Ling. Proc. of the Conference*, pages 133–140.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the Bootstrap*. CRC Press.

Garrido-Alenda, A., Forcada, M. L., and Carrasco, R. C. (2002). Incremental construction and maintenance of morphological analysers based on augmented letter transducers. In *Proc. of Theoretical and Methodological Issues in MT*, pages 53–62.

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 *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. *Procesamiento del Lenguaje Natural*, (43):187–195.

Guzmán, R. (2008). Advanced automatic MT post-editing. *Multilingual Computing*, 19(3):52–57.

Hutchins, W. J. and Somers, H. L. (1992). *An introduction to machine translation*. Academic Press, London.

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

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

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

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Larasati, S. D. and Kuboň, V. (2010). A study of Indonesian-to-Malaysian MT system. In *Proceedings of the 4th International MALINDO Workshop*.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Khudanpur, S., Lane Schwartz, W. T., Weese, J., and Zaidan, O. (2009). Joshua: An open source toolkit for parsingbased machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139. Association for Computational Linguistics.

Mayor, A. and Tyers, F. M. (2009). Matxin: moving towards language independence. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 11–17, Alacant, Spain.

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

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

Phillips, A. B. (2007). Sub-phrasal matching and structural templates in example-based mt. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*.

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.

Sánchez-Cartagena, V. M. and Pérez-Ortiz, J. A. (2010a). ScaleMT: a free/open-source framework for building scalable machine translation web services. *The Prague Bulletin of Mathematical Linguistics*, 93:97–106.

Sánchez-Cartagena, V. M. and Pérez-Ortiz, J. A. (2010b). Tradubi: Open-source social translation for the apertium machine translation platform. *The Prague Bulletin of Mathematical Linguistics*, 93:47–56.

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. (2009). Inferring shallow-transfer machine translation rules from small parallel corpora. *J. of AI Research*, 34:605–635.

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

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.

Scott, B. and Barreiro, A. (2009). Openlogos MT and the SAL representation language. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 19–26.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, "Visions for the Future of Machine Translation"*, pages 223–231, Cambridge, MA, USA.

Thurmair, G. (2009). Comparing different architectures of hybrid machine translation systems. In *Proc. of MT Summit XII*, pages 340—347.

Tyers, F. M. and Alperen, M. S. (2010). SETimes: A parallel corpus of Balkan languages. In *Proceedings of the MultiLR Workshop at the Language Resources and Evaluation Conference, LREC2010*, pages 49–53.

Tyers, F. M. 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., 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.

Way, A. (2010). *The Handbook of Computational Linguistics and Natural Language Processing*, chapter Machine Translation, pages 531–573. Wiley-Blackwell, Oxford, UK.