# Using Apertium linguistic data for tokenization to improve Moses SMT performance

**Sergio Ortiz Rojas, Santiago Cortés Vaíllo**
Prompsit Language Engineering
Campus UMH, Edficio Quorum III
Avda Universitat s/n, E-03202 Elx, Spain
{sergio,santiago}@prompsit.com

## Abstract

This paper describes a new method to tokenize texts, both to train a Moses SMT system and to be used during the translation process. The new method involves reusing the morphological analyser and part-of-speech tagger of the Apertium rule-based machine translation system to enrich the default tokenization used in Moses with part-of-speech-based truecasing, multi-word-unit chunking, number preprocessing and fixed translation patterns. Figures of the experimental results show an improvement of the final quality similar to the improvement attained by using minimum-error-rate training (MERT) as well as an increase of the overall consistency of the output.

## 1 Introduction

Apertium (Tyers et al., 2009) is a free/open-source machine translation (MT) platform that provides rule-based MT (RBMT) systems for an increasing number of languages. Apertium uses human-built linguistic data, consisting of dictionaries (both monolingual for morphological analysis and generation, and bilingual for translation purposes) and transfer rules intended to perform transformations involving more than a translation unit. These data are available for more than 30 different languages, in different degrees of development.

Considering that Apertium is evolving as an independent MT solution, one open question is how other systems, in particular statistical machine translation (SMT) systems, could possibly benefit from Apertium freely available linguistic data. Some researchers have explored hybrid approaches to leverage these Apertium data. Some use the whole Apertium system as a wrapper, managing translations from other engines, like Sánchez-Martínez et al. (2009). Others like Sánchez-Cartagena et al. (2011) use transfer rules and dictionary information to generate translation hypotheses.

In this work we explore a strategy to exploit some of the information stored in the Apertium dictionaries and the part-of-speech tagger of Apertium. On the one hand, the morphological tags deliverd by the part-of-speech tagger of Apertium will be used to decide when to lowercase the first word of a sentence and to split sentences. On the other hand, Apertium dictionaries have translation-oriented multiword-units (MWUs), coded by linguists. We will use MWU information both in the training corpus and during the translation process, as we expect it to provide better alignments (and therefore, better translation quality) during training. The reason which leads us to expect this improvement is that words are particularized using their context and therefore this avoids frequency interferences between words when they appear in MWUs and the individual words that form those MWUs.

During this work we found the experimental fact that using word division determined by linguistically motivated, human encoded data (in our case, from the Apertium platform), can improve consistently SMT quality in all of our experiments.

Factored translation models (Koehn and Hoang, 2007) use equivalent linguistic data without multiword translation units. It uses however a different strategy based on training and using separate trans-

lation models for lemmas and for parts of speech.

We also include in our proposal a way to manage fixed translations, based on Apertium morphology modules, that allow a more stable handling of numbers, some punctuation marks and fixed translations of proper nouns during translation.

In the following sections we detail the specifics of the tokenizing method proposed (section 2), the experiments carried out to evaluate its performance, the results (section 3) and, finally, conclusions and a description of future work (section 4).

## 2 Enriching text tokenization with linguistic data

### 2.1 Baseline: the default tokenization of text in Moses

The default training in Moses is based on texts tokenized in a very crude way: separating words and punctuation, and possibly taking into account some (language-dependent) abbreviations that contain punctuation marks inside. The input texts are lowercased by default and then a *recaser* is trained to attempt to restore the original casing (capitalization) of the text, taking into account the diferent casing in both languages considered in the translation. In the figure 1, in the baseline row, we see how this work is done with an example that will be used throughout this paper.

An alternative way of tokenizing text in Moses, not considered in this work, is using *truecasing*. Truecasing consists in retaining the original case of words but lowercasing only first words of sentences if their most frequent form in texts is lowercased.

### 2.2 Adding Apertium-based tokenization

We use MWUs for sequences of words that are worth to be considered together rather than separately for a particular purpose. In a similar way, we define multiword translation units (MTUs) as translation units that have MWUs in at least on one of the two languages involved.

The components of Apertium being used for this purpose are the morphological analyzer and the part-of-speech tagger. The morphological analyzer is a module based on finite-state technology; it provides all the possible morphological analyses for a given lexical unit, while also tokenizing the input according to the definition of these lexical units in dictionaries (left-to-right, longest-match, tokenize-as-you-analyse strategy). The part-of-speech tagger uses hidden Markov model (HMM) techniques to determine the best part-of-speech of a given word in its context.

The morphological analyzer of Apertium (`lt-proc -a`) marks the start each lexical unit it recognizes with a circumflex sign (`"^"`) and its end with a dollar sign (`"$"`). MWUs are marked together as if it were regular words, including the blank characters found between individual words. The surface form comes first, and then, the different analyses are written, with the bar character (`"/"`) used as a separator.

The part-of-speech tagger (`apertium-tagger -g -p`) uses a suitably-trained HMM to select the most likely part-of-speech tag (and therefore the most likely analysis) among those provided by the morphological analyser.

In order to allow Moses to use this segmentation, blanks inside MWUs are replaced with tilde (`"~"`) characters. The aim of this preprocessing is to reduce the probability of possible relationships between words identified by automated text alignment process that have not been taken into account in order to properly align a bitext when training a SMT model.

Each multiword unit, in this experiment, is not intended to have a matching multiword in the other side. Multiword units are treated as regular words and the alignment process will decide which correspondence applies for every sentence having the perspective that the SMT engine will decide the most likely translation in each case.

Figure 1 shows the result of tokenizing text using Apertium part-of-speech is shown. Particular part-of-speech tags are used in order to decide whether the first word of a sentence has to be lowercased or not, rather than using the frequency of the word in the text as it is done in truecasing.

Figure 2 shows an example of a parallel sentence of the kind used to train the system. The co-occurrence, in this particular case, of Apertium MWUs gives an idea of how can the specifics of tokenization can affect alignment quality and, therefore the translation quality obtained from the trained models.

| | |
|---|---|
| **Original** | A few months ago, the new CEO of Air Berlin, Stephane Richard, announced that the company will base 30% of bonuses on the ”happiness” of their staff. . |
| **Baseline** | a few months ago , the new ceo of air berlin , stephane richard , announced that the company will base 30 % of bonuses on the " happiness " of their staff . |
| **Combined** | **a˜few** months ago , the new **CEO** of **Air˜Berlin , Stephane Richard** , announced that the company will base **_NUM2_%** of bonuses on the **˜"** happiness **"˜** of their staff . |

Figure 1: Combined tokenization using Apertium linguistic data. Note the tilde marks grouping multi-word units, and the preservation of the original casing in “Air Berlin”.

| | **English** | **Spanish** |
|---|---|---|
| **Baseline** | we european socialists are in favour of a market economy with a social purpose . | nosotros , los socialistas europeos , estamos a favor de una economía de mercado con fines sociales . |
| **Combined** | we European Socialists are **in˜favour** of a **market˜economy** with a social purpose . | nosotros , los socialistas europeos , estamos **a˜favor** de una **economía˜de˜mercado** con fines sociales . |

Figure 2: Example of co-occurrence of multiwords in both sides of the training corpus.

### 2.3 Number preprocessing

Statistical machine translators treat numbers as it were usual words. In general, users do not expect numbers to be deeply transformed as a result of MT processing. They might however require some minor transformations such as those affecting the use of punctuation. For example, the English number 2,345.45 should be written in Spanish as 2.345,45 (with dot and comma reversed), following the conventions of the language.

SMT systems do not deal very well with numbers. Numbers are treated like different words and stored in phrase tables and language models. This representation is not suitable since numbers constitute a regular language that can be perfectly characterized by a regular expression. This fact leads to an enormous variability in the training corpora of SMT systems regardless of the number nature and meaning.

For example, years are generally 2 or 4-digit numbers and temperatures 1, 2 or 3-digit numbers, depending on the particular context of a text. We use a transformation mechanism that tries to keep these facts in mind in order to reduce text complexity while mantaining these differences. Numeric sequences are therefore transformed in a input text into the following entities:

- _NUMZ_: represents the 0 number only when occurs as a 1-digit number.
- _NUMI_: represents the 1 number only when occurs as a 1-digit number.
- _NUM[0-9]+_: represents the rest of sequences of numbers, while the number after _NUM indicates the number of digits found.

The specific treatment of numbers 0 and 1 is done in order to reflect the fact that, not only, but these two numbers are treated specially depending on the language. For example, number 1 appears in singular linguistic contexts and make sense to differentiate it from other 1-digit numbers, while 0 is usually followed by plural forms.

A mapping between these entities and the original numbers is stored in a way that it can be retrieved

after running the SMT systems to restore the desired format of the numbers in suitable positions.

An example of this rewriting process is shown in table 1.

| Original | Transformed |
|---|---|
| 2004 | _NUM4_ |
| 2,004 | _NUM1_,_NUM3_ |
| 0.34 | _NUMZ_._NUM2_ |
| 3 000 | _NUM1_ _NUM3_ |
| 0.1 | _NUMZ_._NUMI_ |

Table 1: Some examples of number rewriting.

### 2.4 Fixed translations

Some inconsistencies appearing in the translations generated by Moses are related to missing or altered proper nouns, punctuation marks, numbers and other kind of fixed-translation patterns or fixed-translation entities.

Taking advantage of the Moses XML Markup feature to indicate fixed translations to the decoder and of the information of the Apertium dictionaries, a module to preprocess patterns and rules has been set up to be used during the translation process.

The main advantage of operating this way is the possibility of having consistent translations of well known expressions without requiring large amount of data containing these expressions even when these expressions are not frequent in the training corpora, and also to fix frequent multiword translations determined by linguists.

The module uses Apertium-like dictionaries with both language-dependent and independent data to mark fixed translations and expressions in the source language that will be forced in the target language. These dictionaries are compiled by the Apertium `lt-comp` program to be turned into finite-state-transducers which can be processed at high-speed by the Apertium `lt-proc` program.

A typical fixed translation dictionary contains:

- list of persons, places and entities that do not have to be translated (*Bush*, *Colorado*, *France Telecom*)
- regular expressions for punctuation marks (exclamation marks, quotes, brackets, etc.)

```
<e>
  <p>
    <l>France<b/>Telecom</l>
    <r>France<b/>Telecom</r>
  </p>
</e>
```

Figure 3: Example of an entry in the fixed translation dictionary to avoid *France* to be translated in isolation from English to Spanish or from French to Spanish when it is part of *France Telecom*.

- regular expressions for numerical entities (dates, amounts of money, decimals, percentages, etc.)
- special characters (currency symbols, ampersands, hash marks, etc.)
- regular expressions for URLs
- regular expressions for e-mail addresses

An entry in the dictionary looks like in the example in figure 3.

An example of fixed translation and the way it is marked in the source language text is provided in figure 4. In this case, *Air Berlin* was marked both as a MWU during the tokenization process according to Apertium morphological analysis and as a fixed translation according to the fixed translation dictionary described in this section. Depending on the training corpora, if *Air Berlin* were not a fixed translation, Moses could try to translate *Air* into the target language.

## 3 Experiment and results

In order to evaluate the performance of the new tokenizing method compared to the default tokenizer in Moses, the following experiment has been performed:

- **Baseline**: four baseline systems for English–Spanish, Spanish–English, French–Spanish, Spanish–French have been trained using the WMT11[1] baseline system data (Europarl only), instructions and parameters.

[1] `http://www.statmt.org/wmt11/`

```
<fixed-translation translation="Air˜Berlin">Air˜Berlin</fixed-translation>
```

Figure 4: Air Berlin is marked using Moses XML Markup feature according to Apertium morphological analysis and fixed translations dictionary.

- **Combined**: systems for the same four language pairs have been trained following the same procedure but with by tokenizing the same training data using the method described in this paper.

The experiment have been carried out using WMT11 baseline system data, instructions and parameters to train baseline models, and then replacing the data by the same data tokenized in the way exposed in this paper. The test set corresponds also to WMT11 task, 2500 sentences from NewsCommentary 2010.

Results for BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and NIST (Doddington, 2002) scores are presented in tables 2, 3 and 4, respectively. The figures show that, for all four systems, those trained with the new tokenizer method outperform the baseline systems. In all cases, at least 0.02 BLEU points are gained in the combined systems. METEOR shows a better improvement in the French–Spanish system than in the other three, which show improvements between 0.01 and 0.02. NIST scores also show a general improvement for all combined systems.

| Translator | Baseline | Combined |
|---|---|---|
| fr → es | 0.27 | 0.29 |
| es → fr | 0.25 | 0.27 |
| en → es | 0.22 | 0.24 |
| es → en | 0.22 | 0.24 |

Table 2: BLEU scores for the experiments

This linguistic-motivated tokenizing method proves to be useful to increase the final quality of the translation by making it more consistent with respect to casing, punctuation and other fixed patterns.

| Translator | Baseline | Combined |
|---|---|---|
| fr → es | 0.42 | 0.45 |
| es → fr | 0.40 | 0.41 |
| en → es | 0.38 | 0.40 |
| es → en | 0.24 | 0.26 |

Table 3: METEOR scores for the experiments

| Translator | Baseline | Combined |
|---|---|---|
| fr → es | 7.22 | 7.55 |
| es → fr | 6.90 | 7.21 |
| en → es | 6.49 | 6.95 |
| es → en | 6.62 | 7.02 |

Table 4: NIST scores for the experiments

## 4 Conclusions and future work

A new linguistic-based tokenization method to preprocess the texts that are used to train a Moses SMT system has been presented in this paper; the method uses the linguistic data freely available in the Apertium project. This way of combining RBMT resources with SMT has shown to improve SMT results consistently as measured with the standard metrics. The availability of data in the Apertium platform and from other sources makes possible to apply this method to a variety of languages. Additionally, this processing does not conflict with other techniques that may be applied to further improve SMT quality.

In the future, we will continue to explore ways of integration between Apertium and Moses at a deeper level, in order to make this first Apertium–Moses combined system more stable and reliable, in order to obtain a significant improvement in the output translation quality.

Some of the improvements could come from using linguistic information to reorder some sequences of parts of speech between languages with large structural differences or from filtering training cor-

pora using dictionary equivalences in order to remove very unfrequent translations.

## 5 Acknowledgments

We wish to thank Gema Ramírez Sánchez and professor Mikel L. Forcada for their support during the writing of this paper.

## References

Francis M. Tyers and Mikel L. Forcada and Gema Ramírez-Sánchez. 2009. *The Apertium machine translation platform: Five years on*. Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, pages 3–10, Alacant, Spain.

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

Víctor M. Sánchez-Cartagena and Felipe Sánchez-Martínez and Juan A. Pérez-Ortiz. 2011. *The Universitat d'Alacant hybrid machine translation system for WMT2011*. Proceedings of the 6th Workshop on Statistical Machine Translation, pages 456–463, Edinburgh, United Kingdom.

Philipp Koehn and Hieu Hoang. 2007. *Factored Translation Models*. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 868–876., Praque, Czech Republic.

Philipp Koehn and Hieu Hoang and Alexandra Birch and Chris Callison-Burch and Marcello Federico and Nicola Bertoldi and Brooke Cowan and Wade Shen and Christine Moran and Richard Zens and Chris Dyer and Ondrej Bojar and Alexandra Constantin and Evan Herbst. 2007. *Moses: Open source toolkit for statistical machine translation*. Proceedings of the ACL 2007 Demo and Poster Sessions, pages 177–180, Prague, Czech Republic.

Kishore Papineni and Salim Roukos and Todd Ward and Wei-Jing Zhu. 2002. *Bleu: a method for automatic evaluation of machine translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*. Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan.

George Doddington. 2002. *Automatic evaluation of machine translation quality using n-gram co-occurrence statistics*. Proceedings of the 2nd international conference on Human Language Technology Research, pages 138–145, San Diego, California, USA.