# Speeding up Target-Language Driven Part-of-Speech Tagger Training for Machine Translation

Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, and Mikel L. Forcada

Transducens Group – Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant, Spain
{fsanchez,japerez,mlf}@dlsi.ua.es

**Abstract.** When training hidden-Markov-model-based part-of-speech (PoS) taggers involved in machine translation systems in an unsupervised manner the use of target-language information has proven to give better results than the standard Baum-Welch algorithm. The target-language-driven training algorithm proceeds by translating every possible PoS tag sequence resulting from the disambiguation of the words in each source-language text segment into the target language, and using a target-language model to estimate the likelihood of the translation of each possible disambiguation. The main disadvantage of this method is that the number of translations to perform grows exponentially with segment length, translation being the most time-consuming task. In this paper, we present a method that uses *a priori* knowledge obtained in an unsupervised manner to prune unlikely disambiguations in each text segment, so that the number of translations to be performed during training is reduced. The experimental results show that this new pruning method drastically reduces the amount of translations done during training (and, consequently, the time complexity of the algorithm) without degrading the tagging accuracy achieved.

## 1 Introduction

One of the classical ways to train part-of-speech (PoS) taggers based on hidden-Markov-models [1] (HMM) in an unsupervised manner is by means of the Baum-Welch algorithm [2]. However, when the resulting PoS tagger is to be embedded within a machine translation (MT) systems, the use of information not only from the source language (SL), but also from the target language (TL) has proven to give better results [3].

The TL-driven training algorithm [3] proceeds by translating every possible PoS tag sequence resulting from the disambiguation of the words in each SL text segment into the TL, and using a probabilistic TL model to estimate the likelihood of the translation corresponding to each possible disambiguation. The main disadvantage of this method is that the number of possible disambiguations to translate grows exponentially with the segment length. As a consequence of that, segment length must be constrained to keep time complexity under control,

therefore rejecting the potential benefits of likelihoods estimated from longer segments. Moreover, translation is the most time-consuming task of the training algorithm.

This paper presents a method that uses *a priori* knowledge, obtained in an unsupervised manner, to prune or rule out unlikely disambiguations of each segment, so that the number of translations to be performed is reduced. The method proceeds as follows; first, the SL training corpus is preprocessed to compute initial HMM parameters; and then, the SL corpus is processed by the TL-driven training algorithm using the initial HMM parameters to prune, that is, to avoid translating the least likely disambiguations of each SL text segment. The experimental results show that the number of words to be translated by the TL-driven training algorithm is drastically reduced without degrading the tagging accuracy. Moreover, we have found out that the tagging accuracy is slightly better when pruning.

As seen in section 5, the open-source MT engine Opentrad Apertium [4], which uses HMM-based PoS tagging during SL analysis, has been used for the experiments. It must be pointed out that the TL-driven training method described in [3], along with the pruning method proposed in this paper, have been implemented in the package name `apertium-tagger-training-tools`, and released under the GPL license.[1]

The rest of the paper is organized as follows: Section 2 overviews the use of HMM for PoS tagging. In section 3 the TL-driven HMM-based PoS tagger training method is explained; then, in section 4 the pruning technique used in the experiments is explained in detail. Section 5 overviews the open-source MT engine used to test our new approach, the experiments conducted and the results achieved. Finally, in sections 6 and 7 the results are discussed and future work is outlined.

## 2 Hidden Markov Models for Part-of-Speech Tagging

This section overviews the application of HMMs in the natural language processing field as PoS taggers.

A first-order HMM [1] is defined as $\lambda = (\Gamma, \Sigma, A, B, \pi)$, where $\Gamma$ is the set of states, $\Sigma$ is the set of observable outputs, $A$ is the $|\Gamma|\times|\Gamma|$ matrix of state-to-state transition probabilities, $B$ is the $|\Gamma|\times|\Sigma|$ matrix with the probability of each observable output $\sigma \in \Sigma$ being emitted from each state $\gamma \in \Gamma$, and the vector $\pi$, with dimensionality $|\Gamma|$, defines the initial probability of each state. The system produces an output each time a state is reached after a transition.

When a first-order HMM is used to perform PoS tagging, each HMM state $\gamma$ is made to correspond to a different PoS tag, and the set of observable outputs $\Sigma$ are made to correspond to *word classes*. In many applications a word class is an *ambiguity class* [5], that is, the set of all possible PoS tags that a word could receive. Moreover, when a HMM is used to perform PoS tagging, the estimation

[1] The MT engine and the `apertium-tagger-training-tools` package can be downloaded from `http://apertium.sourceforge.net`.

of the initial probability of each state can be conveniently avoided by assuming that each sentence begins with the end-of-sentence mark. In this case, $\pi(\gamma)$ is 1 when $\gamma$ is the end-of-sentence mark, and 0 otherwise. A deeper description of the use of this kind of statistical models for PoS tagging may be found in [5] and [6, ch. 9].

## 3 Target-Language-Driven Training Overview

This section overviews the TL-driven training method that constitutes the basis of the work reported in this paper. A deeper and more formal description of the TL-driven training method may be found in [3].

Typically, the training of HMM-based PoS taggers is done using the *maximum-likelihood estimate* (MLE) method [7] when tagged corpora[2] are available (supervised method), or using the Baum-Welch algorithm [2,5] with untagged corpora[3] (unsupervised method). But when the resulting PoS tagger is to be embedded as a module of a working MT system, HMM training can be done in an unsupervised manner by using information not only from the SL, but also from the TL.

The main idea behind the use of TL information is that the correct disambiguation (tag assignment) of a given SL segment will produce a more likely TL translation than any (or most) of the remaining wrong disambiguations. In order to apply this method these steps are followed: first the SL text is segmented; then, the set of all possible disambiguations for each text segment are generated and translated into the TL; next, a statistical TL model is used to compute the likelihood of the translation of each disambiguation; and, finally, these likelihoods are used to adjust the parameters of the SL HMM: the higher the likelihood, the higher the probability of the original SL tag sequence in the HMM being trained. The number of possible disambiguations of a text segment grows exponentially with its length; therefore, the number of translations to be performed by this training algorithm is very high. Indeed, the translation of segments is the most time-consuming task in this method.

Let us illustrate how this training method works with the following example. Consider the following segment in English, $s =$ *"He books the room"*, and that an indirect MT system translating between English and Spanish is available. The first step is to use a morphological analyzer to obtain the set of all possible PoS tags for each word. Suppose that the morphological analysis of the previous segment according to the lexicon is: *He* (pronoun), *books* (verb or noun), *the* (article) and *room* (verb or noun). As there are two ambiguous words (*books* and *room*) we have, for the given segment, four disambiguation *paths* or PoS combinations, that is to say:

[2] In a *tagged corpus* each occurrence of each word (ambiguous or not) has been assigned the correct PoS tag.

[3] In an *untagged corpus* all words are assigned (using, for instance, a morphological analyzer) the set of all possible PoS tags independently of context.

- $\boldsymbol{g_1}$ = (pronoun, verb, article, noun),
- $\boldsymbol{g_2}$ = (pronoun, verb, article, verb),
- $\boldsymbol{g_3}$ = (pronoun, noun, article, noun), and
- $\boldsymbol{g_4}$ = (pronoun, noun, article, verb).

Let $\tau$ be the function representing the translation task. The next step is to translate the SL segment into the TL according to each disambiguation path $\boldsymbol{g_i}$:

- $\tau(\boldsymbol{g_1}, s)$ = "*Él reserva la habitación*",
- $\tau(\boldsymbol{g_2}, s)$ ="*Él reserva la aloja*",
- $\tau(\boldsymbol{g_3}, s)$ ="*Él libros la habitación*", and
- $\tau(\boldsymbol{g_4}, s)$ ="*Él libros la aloja*".

It is expected that a Spanish language model will assign a higher likelihood to translation $\tau(\boldsymbol{g_1}, s)$ than to the other ones, which make little sense in Spanish. So the tag sequence $\boldsymbol{g_1}$ will have a higher probability than the other ones.

To estimate the HMM parameters, the calculated probabilities are used as if fractional counts were available to a supervised training method based on the MLE method in conjunction with a smoothing technique. In the experiments reported in section 5 to estimate the HMM parameters we used the *expected likelihood estimate* (ELE) method [7] that consists of adding a fixed initial count to each event before applying the MLE method.

## 4 Pruning of Disambiguation Paths

Next, we focus on the main disadvantage of this training method (the large number of translations that need to be performed) and how to overcome it. The aim of the new method presented in this section is to reduce as much as possible the number of translations to perform without degrading the tagging accuracy achieved by the resulting PoS tagger.

### 4.1 Pruning Method

The disambiguation pruning method is based on *a priori* knowledge, that is, on an initial model $M_{\text{tag}}$ of SL tags. The assumption here is that any reasonable model of SL tags may prove helpful to choose a set of possible disambiguation paths, so that the correct one is in that set. Therefore, there is no need to translate all possible disambiguation paths of each segment into the TL, but only the most "promising" ones.

The model $M_{\text{tag}}$ of SL tags to be used can be either a HMM or another model whose parameters are obtained by means of a statistically sound method. Nevertheless, using a HMM as an initial model allows the method to dynamically update it with the new evidence collected during training (see section 4.2 for more details).

The pruning of disambiguation paths for a given SL text segment $s$ is carried out as follows: First, the *a priori* likelihood $p(\boldsymbol{g_i}|s, M_{\text{tag}})$ of each possible disambiguation path $\boldsymbol{g_i}$ of segment $s$ is calculated given the tagging model $M_{\text{tag}}$; then,

the set of disambiguation paths to take into account is determined according to the calculated *a priori* likelihoods.

Let $T(s) = \{\boldsymbol{g_1}, \ldots, \boldsymbol{g_n}\}$ be the set of all possible disambiguation paths of SL segment $s$, ordered in decreasing order of their *a priori* likelihood $p(\boldsymbol{g_i}|s, M_{\text{tag}})$. To decide which disambiguation paths to take into account, the pruning algorithm is provided with a mass probability threshold $\rho$. Thus, the pruning method takes into account only the most likely disambiguation paths of $T(s)$ that make the probability mass threshold $\rho$ to be reached. Therefore, for each segment $s$ the subset $T'(s) \subseteq T(s)$ of disambiguation paths finally taken into account satisfies the property

$$\rho \leq \sum_{\forall \boldsymbol{g_i} \in T'(s)} p(\boldsymbol{g_i}|s, M_{\text{tag}}). \tag{1}$$

### 4.2 HMM Updating

This section explains how the model used for pruning can be updated during training so that it integrates new evidence collected from the TL. The idea is to periodically estimate a HMM using the counts collected from the TL (as explained in section 3), and to mix the resulting HMM with the initial one; the mixed HMM becomes the new model $M_{\text{tag}}$ used for pruning.

The initial model and an improved model obtained during training are mixed so that *a priori* likelihoods are better estimated. The mixing consists, on the one hand of the mixing of the transition probabilities $a_{\gamma_i\gamma_j}$ between HMM states; and, on the other hand, of the mixing of the emission probabilities $b_{\gamma_i\sigma_k}$ of each observable output $\sigma_k$ being emitted from each HMM state $\gamma_i$.

Let $\boldsymbol{\theta} = (a_{\gamma_1\gamma_1}, ..., a_{\gamma_{|\Gamma|}\gamma_{|\Gamma|}}, b_{\gamma_1\sigma_1}, ..., b_{\gamma_{|\Gamma|}\sigma_{|\Sigma|}})$ be a vector containing all the parameters of a given HMM. The mixing of the initial HMM and the new one can be done through the next equation:

$$\boldsymbol{\theta}_{\mathbf{mixed}}(x) = \lambda(x)\, \boldsymbol{\theta}_{\mathbf{TL}}(x) + (1 - \lambda(x))\, \boldsymbol{\theta}_{\mathbf{init}}, \tag{2}$$

where $\boldsymbol{\theta}_{\mathbf{mixed}}(x)$ refers to the HMM parameters after mixing the two models when $x$ words of the training corpus have been processed; $\boldsymbol{\theta}_{\mathbf{TL}}(x)$ refers to the HMM parameters estimated by means of the TL-driven method after processing $x$ words of the SL training corpus; and $\boldsymbol{\theta}_{\mathbf{init}}$ refers to the parameters of the initial HMM. Function $\lambda(x)$ assigns a weight to the model estimated using the counts collected from the TL ($\boldsymbol{\theta}_{\mathbf{TL}}$). This weight function is made to depend on the number $x$ of SL words processed so far. This way the weight of each model can be changed during training.

## 5 Experiments

In this section we overview the MT system used to train the PoS tagger by means of the TL-driven training algorithm, the experiments conducted, and the results achieved.

### 5.1 Machine Translation Engine

This section introduces the MT system used in the experiments, although almost any other MT architecture (which uses a HMM-based PoS tagger) may also be used in combination with the TL-driven training algorithm.

We used the open-source shallow-transfer MT engine Opentrad Apertium [4,8] together with linguistic data for the Spanish–Catalan language pair.[4] This MT engine follows a shallow transfer approach and consists of the following pipelined modules:

- A *morphological analyzer* which tokenizes the text in surface forms (SF) and delivers, for each SF, one or more lexical forms (LF) consisting of *lemma*, *lexical category* and morphological inflection information.
- A *PoS tagger* which chooses, using a first order HMM as described in section 2, one of the LFs corresponding to an ambiguous SF. This is the module whose training is considered in this paper.
- A *lexical transfer* module which reads each SL LF and delivers the corresponding TL LF.
- A *structural transfer* module (parallel to the lexical transfer) which uses a finite-state chunker to detect patterns of LFs which need to be processed for word reorderings, agreement, etc. and performs these operations.
- A *morphological generator* which delivers a TL SF for each TL LF, by suitably inflecting it, and performs other orthographical transformations such as contractions.

### 5.2 Results

We have tested the approach presented in section 4 to train a HMM-based PoS tagger for Spanish, being Catalan the TL, through the MT system described above.

As mentioned in section 3, the TL-driven training method needs a TL model to score the different translations $\tau(\boldsymbol{g_i}, s)$ of each SL text segment $s$. In this paper we have used a classical trigram language model like the one used in [3]. This language model was trained on a raw-text Catalan corpus with around 2 000 000 words.

To study the behaviour of our pruning method, experiments have been performed with 5 disjoint SL (Spanish) corpora of 500 000 words each. With all the corpora we proceeded in the same way: First the initial model was computed by means of Kupiec's method [9], a common unsupervised initialization method often used before training HMM-based PoS taggers through the Baum-Welch algorithm. After that, the HMM-based PoS tagger was trained by means of the TL-driven training method described in section 3. The HMM used for pruning was updated after every 1 000 words processed as explained in section 4.2. To

[4] Both the MT engine and the linguistic data used can be downloaded from `http://apertium.sourceforge.net`. For the experiments we have used the packages `lltoolbox-1.0.1`, `apertium-1.0.1` and `apertium-es-ca-1.0.1`

this end, the weighting function $\lambda(x)$ used in equation (2) was chosen to grow linearly from 0 to 1 with the amount $x$ of words processed:

$$\lambda(x) = x/C, \tag{3}$$

where $C = 500\,000$ is the total number of words of the SL training corpus.

In order to determine the appropriate mass probability threshold $\rho$ that speeds the TL-driven method up without degrading its PoS tagging accuracy we considered a set of values for $\rho$ between 0.1 and 1.0 at increments of 0.1. Note that when $\rho = 1.0$, no pruning is done; that is, all possible disambiguation paths of each segment are translated into the TL.

Figure 1 shows, for three different values of the probability mass threshold $\rho$, the evolution of the mean and the standard deviation of the PoS tagging error rate; in all cases the HMM being evaluated is the one used for pruning. The error rates reported are measured on a representative Spanish (SL) tagged corpus with around 8 000 words, and are calculated over ambiguous and unknown words only, not over all words. The three different values of the probability mass threshold $\rho$ shown are: the smallest threshold used (0.1), the threshold that provides the best PoS tagging performance (0.6, see also figure 2), and the threshold that makes no pruning at all (1.0).

As can be seen in figure 1 the results achieved by the TL-driven training method are better when $\rho = 0.6$ than when $\rho = 1.0$. Convergence is reached earlier when $\rho = 1.0$.

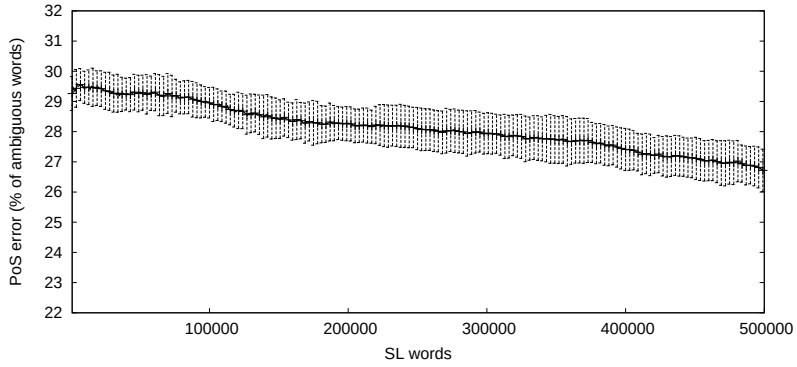
Figure 2 shows the mean and standard deviation of the final PoS tagging error rate achieved after processing the whole training corpora for the different values of $\rho$. As can be seen, the best results are achieved when $\rho = 0.6$, indeed better than the result achieved when no pruning is performed. However, the standard deviation is smaller when no pruning is done ($\rho = 1.0$).

As to how many translations are avoided due to the proposed pruning method, figure 3 shows the average ratio, and standard deviation, of the words finally translated to the total number of words to translate when no pruning is performed. As can be seen, with $\rho = 0.6$ the percentage of words translated is around 16%. This percentage can be seen as roughly proportional to the percentage of disambiguation paths needed to reach the corresponding mass probability threshold.

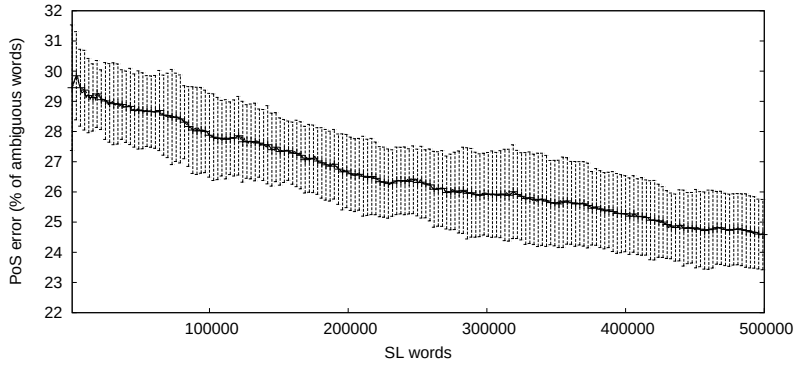
## 6 Discussion

The main disadvantage of the TL-driven method used to train HMM-based PoS taggers [3] is that the number of translations to perform for each SL text segment grows exponentially with the segment length. In this paper we have proposed and tested a new approach to speed up this training method by using *a priori* knowledge obtained in an unsupervised way from the SL.

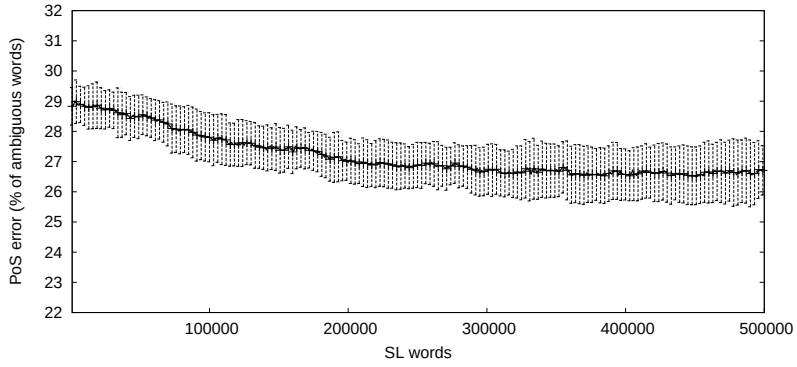
The method proposed consists of pruning the most unlikely disambiguation paths (PoS combinations) of each SL text segment processed by the algorithm. This pruning method is based on the assumption that any reasonable model of




(a) $\rho = 0.1$




(b) $\rho = 0.6$




(c) $\rho = 1.0$

**Fig. 1:** Mean and standard deviation of the PoS tagging error rate for three different values of the probability mass threshold $\rho$ depending on the number of words processed by the training algorithm. The error rates reported are measured using a Spanish (SL) tagged corpus with around 8 000 words, and are calculated over ambiguous and unknown words only, not over all words.

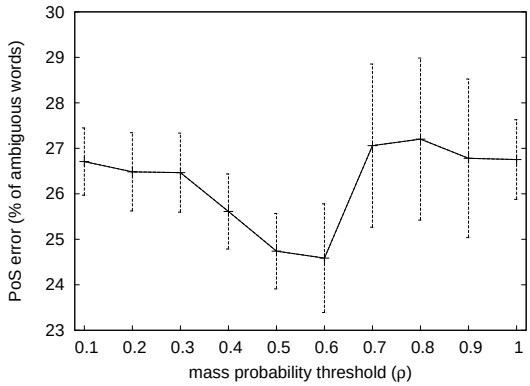



**Fig. 2:** Mean and standard deviation of the final PoS tagging error rate achieved after processing the whole corpus of 500 000 words for the different values of $\rho$ used.

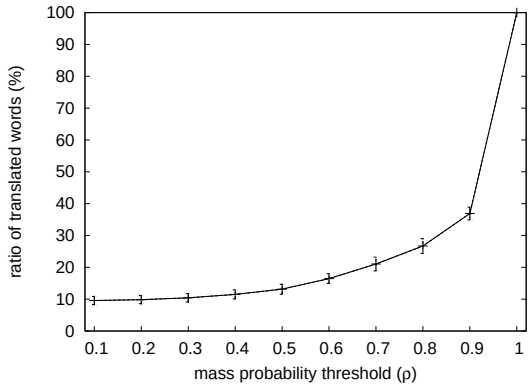



**Fig. 3:** Percentage of translated words for each value of the probability mass threshold $\rho$. The percentage of translated words is calculated over the total number of words that are translated when no pruning is done.

SL tags may prove helpful to choose a set of possible disambiguation paths, the correct one being included in that set. Moreover, the model used for pruning can be updated along the training with the new data collected while training.

The method presented has been tested on five different corpora and with different mass probability thresholds. The results reported in section 5.2 show, on the one hand, that the pruning method described avoids more than 80% of the translations to perform; and on the other hand, that the results achieved by the TL-driven training method improve if improbable disambiguation paths are not taken into account. This could be explained by the fact that HMM parameters associated to discarded disambiguation paths have a null count; however, when no pruning is done their TL-estimated fractional counts are small, but never null.

## 7 Future Work

The pruning method described is based on *a priori* knowledge used to calculate the *a priori* likelihood of each possible disambiguation path of a given SL text segment. It has been explained how to update the model used for pruning by mixing the initial HMM provided to the algorithm with the HMM calculated from the counts collected from the TL. In the experiments reported both HMMs have been mixed through equation (2), which needs to be provided with a weighting function $\lambda$.

For the experiments we have used the simplest possible weighting function (see equation (3)). This function makes the initial model provided to the algorithm to have a higher weight than the model being learned from the TL until one half of the SL training corpus is processed. In order to explore how fast does the TL-driven training method learns, we plan to try other weighting functions giving earlier a higher weight to the model being learned from the TL.

Finally, we want to test two additional strategies to select the set of disambiguation paths to take into account; on the one hand, a method that changes the probability mass threshold along the training; and on the other hand, a method that instead of using a probability mass threshold uses a fixed number of disambiguation paths ($k$-best). The last one could be implemented in such a way in which all *a priori* likelihoods do not need to be explicitly calculated before discarding many of them.

## Acknowledgements

Work funded by the Spanish Ministry of Science and Technology through project TIC2003-08681-C02-01, and by the Spanish Ministry of Education and Science and the European Social Found through research grant BES-2004-4711. We thank Rafael C. Carrasco (Universitat d'Alacant, Spain) for useful comments on the target-language-driven training method.

## References

1. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2) (1989) 257–286
2. Baum, L.E.: An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. Inequalities **3** (1972) 1–8
3. Sánchez-Martínez, F., Pérez-Ortiz, J.A., Forcada, M.L.: Exploring the use of target-language information to train the part-of-speech tagger of machine translation systems. In: Advances in Natural Language Processing, Proceedings of 4th International Conference EsTAL. Volume 3230 of Lecture Notes in Computer Science. Springer-Verlag (2004) 137–148
4. Corbí-Bellot, A.M., Forcada, M.L., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Alegria, I., Mayor, A., Sarasola, K.: An open-source shallow-transfer machine translation engine for the Romance languages of Spain. In: Proceedings of the 10th European Associtation for Machine Translation Conference, Budapest, Hungary (2005) 79–86

5. Cutting, D., Kupiec, J., Pedersen, J., Sibun, P.: A practical part-of-speech tagger. In: Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference., Trento, Italia (1992) 133–140
6. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT press (1999)
7. Gale, W.A., Church, K.W.: Poor estimates of context are worse than none. In: Proceedings of a workshop on Speech and natural language, Morgan Kaufmann Publishers Inc. (1990) 283–287
8. Armentano-Oller, C., Carrasco, R.C., Corbí-Bellot, A.M., Forcada, M.L., Ginestí-Rosell, M., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M.A.: Open-source Portuguese-Spanish machine translation. In: 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 (2006) 50–59
9. Kupiec, J.: Robust part-of-speech tagging using a hidden Markov model. Computer Speech and Language **6**(3) (1992) 225–242