Cooperative Unsupervised Training of the Part-of-Speech Taggers in a Bidirectional Machine Translation System

Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada
Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant
E-03071 Alacant, Spain

{fsanchez,japerez,mlf}@dlsi.ua.es

Abstract

When building a machine translation system, the embedded part-of-speech (PoS) tagger deserves special attention, since PoS ambiguities are one of the main sources of mistranslations, specially when related languages are involved. The standard statistical approach for PoS tagging are hidden Markov models (HMM) properly trained by collecting statistics from source-language texts. In the case of bidirectional machine translation systems, this kind of training is often individually performed on each PoS tagger without taking into account the other language, that is, the corresponding target language. But target-language information may help to improve performance. In this paper, a new method is proposed which trains both PoS taggers simultaneously using mutual interaction: at every iteration, the parameters of the HMM corresponding to one of the languages are refined by using the statistical data supplied by the current HMM for the other language. Both models bootstrap by learning cooperatively in an unsupervised manner and require only monolingual texts; no aligned texts are needed. Preliminary results are promising and surpass those of traditional unsupervised approaches.

1 Introduction

One of the main sources of errors in machine translation (MT) systems, specially for related languages, is the incorrect resolution of part-of-speech (PoS) ambiguities. Hidden Markov models (HMMs, Rabiner 1989) are the standard statistical approach (Cutting et al. 1992) to automatic PoS tagging. Typically, unsupervised training of this kind of taggers has been carried out from source-language (SL) untagged corpora (see below in this introduction) using the Baum-Welch algorithm (Rabiner 1989). But target-language (TL) information may also be taken into account in order to improve the performance of these PoS taggers, specially as to the resulting translation quality, an aspect not addressed by training algorithms which use information from the SL only.

Statistics from the TL might be a source of useful information as well, an idea which will be explored in this paper; indeed, using TL information to train a PoS tagger when it is going to be embedded in a MT system is a very natural approach. We propose a new method intended for training simultaneously two PoS taggers for different languages by mutual interaction: the parameters for the HMM-based tagger for each language are alternatively computed from information learned by the other one. The new approach is specially suitable for the bidirectional MT systems including these two taggers, although the resulting taggers may also be useful in other applications in the field of natural language processing.

The proposed method considers the likelihood in the TL of the translation of each of the multiple disambiguations of a source text which can be produced depending on how its PoS ambiguity is resolved. To achieve this goal, these steps are followed: first, the SL text is segmented; then, the set of all possible disambiguations for each segment is generated; after that, each disambiguation is *translated* into TL; next, a TL statistical model is used to compute the likelihood of each translated disambiguation of the segment; 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 model being trained. Rules for text segmentation must be carefully chosen so that the resulting segments are treated independently by the rest of the modules in the MT system.

Our first attempts (Sánchez-Martínez et al. 2004) considered TL *word* trigrams as a TL statistical model and the results were quite promising; in fact, error rates were lower than those obtained using the Baum-Welch algorithm; every possible disambiguation of each SL segment was automatically translated into the TL in order to estimate its likelihood with the TL trigram model. The whole process was unsupervised.

In this paper we follow a completely different approach: we focus on the case of designing bidirectional MT systems and show how to train the two corresponding PoS taggers simultaneously in a single iterative process using cooperative learning. A brief overview of our proposal follows: consider a system translating between languages A and B. At every iteration, the existing HMM for A is refined by using B as TL and estimating the likelihood of the translations of each of the segments in A with the current HMM for B. Then, the roles are interchanged and the existing HMM for B is updated correspondingly. The resulting algorithm is based on a model of TL *tags* —instead of words— and still works in an unsupervised manner.

Yarowsky & Ngai (2001) proposed a method which also uses information from TL in order to train PoS taggers. They consider information from aligned parallel corpora and from (at least) one manually tagged corpus for the TL. Our method, however, needs neither aligned parallel corpora nor manually tagged texts.

Most current MT systems follow the *indirect* or *transfer* approach (Hutchins & Somers 1992, ch. 4): SL text is analysed and converted into an intermediate representation which becomes the basis for generating the corresponding TL text. Analysis modules usually include a PoS tagger for the SL. Our method for training PoS taggers may be applied, in principle, to any variant of an indirect architecture which uses a HMM-based PoS tagger and which includes a morphological generation phase. In particular, a MT system using a classical *morphological transfer* architecture will be considered in the experiments.

We will refer to a text as *unambiguously tagged* or just *tagged* when each occurrence of each word (ambiguous or not) has been assigned the correct PoS tag. An *ambiguously tagged* or *untagged* text corpus is one in which all words are assigned (using a morphological analyser) the set of possible PoS tags independently of context; in this case, ambiguous and unknown words would receive more than one PoS tag (unknown words, that is, words not found in the lexicon, are usually assigned the set of *open* categories, that is, categories which are likely to grow by adding new words of the language: nouns, verbs, adjectives, adverbs and proper nouns). Words receiving the same set of PoS tags are said to belong to the same *ambiguity class* (Cutting et al. 1992); for example, the words *tailor* and *book* both belong to the ambiguity class {noun, verb}.

The paper is organized as follows. Section 2 reviews the basis of the use of HMM in disambiguation tasks and discusses existing methods for PoS tagger training; section 3 describes our general proposal for HMM training based on TL information and the particular approach followed in this paper; section 4 introduces the translation engine and shows the main results of the experiments; finally, in sections 5 and 6 we discuss the results and outline future work to be done.

2 Part-of-Speech Tagging

When a HMM is used for lexical disambiguation purposes in the *ambiguity class* mode (in which each input word is replaced by its corresponding ambiguity class) each HMM state is made to correspond to a different PoS tag and the set of observable items consists of all possible ambiguity classes (Cutting et al. 1992). Building a PoS tagger based on HMMs for the SL in a MT system usually implies:

1. *Designing or adopting a reduced tagset* (set of PoS) which groups the finer tags delivered by the morphological analyser into a small set of coarse tags adequate to the translation task (for example, singular feminine nouns and plural feminine nouns may be grouped under the "noun" tag). Additionally, the number of different lexical probabilities in the HMM is usually drastically reduced by grouping words in ambiguity classes.

2. *Estimating proper HMM parameters*, that is, finding adequate values of the parameters of the model. Existing methods may be grouped according to the kind of corpus they use as input: *supervised* methods require *tagged texts* (see the introduction); *unsupervised* methods are able to extract information from *untagged texts*, that is, from sequences of ambiguity classes.

   On the one hand, estimating parameters from a tagged corpus is usually the best way to improve performance, but tagged corpora are expensive to obtain and require costly human supervision. A supervised method counts the number of occurrences in the corpus of some tag contexts (usually bigrams) and uses this information to determine the values of the parameters of the HMM. On the other hand, for the unsupervised approach no analytical method is known, and existing methods, such as the Baum-Welch algorithm (Rabiner 1989), are only guaranteed to reach to local (not global) maxima of the expectation. We propose a new unsupervised method which requires only untagged SL and TL texts which need not to be aligned and may even be unrelated.

Source language	Target language	$p(g_i \mid s)$
$s \equiv$ y {CNJ} la {ART, PRN} para {VB, PR} si {CNJ}	$g_1 \equiv$ CNJ ART PR CNJ	0.0538
	$g_2 \equiv$ CNJ ART VB CNJ	0.0897
	$g_3 \equiv$ CNJ PRN PR CNJ	0.0027
	$g_4 \equiv$ CNJ PRN VB CNJ	0.8538

Figure 1: Example of an ambiguous SL (Spanish) text segment, paths (in Catalan) resulting from each possible disambiguation, and the estimated likelihood $p(g_i|s)$ for each one.

3 Target-Language-Based Training

3.1 Estimation of a SL HMM from TL information

This section gives mathematical details on how to train a SL HMM using information from the TL.

Let S be the whole SL corpus, s be a (possibly ambiguous) segment from S, g_i a sequence of tags resulting from one of the possible disambiguation choices in s, $\tau(g_i, s)$ the corresponding translation of this g_i in the TL, and $p_{\text{TL}}(\tau(g_i, s))$ the likelihood of $\tau(g_i, s)$ in some TL model. We will call each g_i a *path* since it describes a unique state path in the HMM and write $g_i \in T(s)$ to show that g_i is a possible disambiguation of the words in s. Now, the likelihood of path g_i from segment s may be estimated as:

$$p(g_i|s) = \frac{p(g_i|\tau(g_i, s))\, p_{\text{TL}}(\tau(g_i, s))}{\displaystyle\sum_{g_j \in T(s)} p(g_j|\tau(g_j, s))\, p_{\text{TL}}(\tau(g_j, s))} \tag{1}$$

where the term $p(g_i|\tau(g_i, s))$ is the conditional probability of g_i given translation $\tau(g_i, s)$. That is, the likelihood of path g_i in source segment s is made proportional to the TL likelihood of their translation $\tau(g_i, s)$, but needs to be corrected by a weight $p(g_i|\tau(g_i, s))$, because more than one g_i may contribute to the same $\tau(g_i, s)$. Figure 1 illustrates the use of TL information as already described.

The fact that more than one path in segment s, say without loss of generality g_i and g_j, can produce the same translation in TL (that is, $\tau(g_i, s) = \tau(g_j, s)$ with $i \neq j$) does not imply that $p(g_i|\tau(g_i, s)) = p(g_j|\tau(g_j, s))$. Indeed, the real probabilities of paths are in principle unknown (note that their computation is the main goal of the training method). In the absence of such information, the contributions of each path will be approximated in this paper to be equally likely:

$$p(g_i|\tau(g_i, s)) = \frac{1}{\text{card}\left(\{g_j \in T(s) : \tau(g_j, s) = \tau(g_i, s)\}\right)} \tag{2}$$

Now, we describe how to obtain the parameters of the HMM from the estimated likelihood of each path in each segment, $p(g_i|s)$, which will be treated as a fractional count. An estimate of tag pair occurrence frequency based on $p(g_i|s)$ is:

$$\tilde{n}(\gamma_i \gamma_j) \cong \sum_{s \in S} \sum_{g_i \in T(s)} C_{s,g_i}(\gamma_i, \gamma_j)\, p(g_i|s) \tag{3}$$

where $C_{s,g_i}(\gamma_i, \gamma_j)$ is the number of times tag γ_i is followed by tag γ_j in path g_i of segment s. Therefore, the HMM parameter $a_{\gamma_i\gamma_j}$ corresponding to the transition probability from the state associated with tag γ_i to the state associated with tag γ_j (Rabiner 1989; Cutting et al. 1992) can be computed as follows:

$$a_{\gamma_i\gamma_j} = \frac{\tilde{n}(\gamma_i\gamma_j)}{\sum_{\gamma_k\in\Gamma}\tilde{n}(\gamma_i\gamma_k)} \quad (4)$$

where Γ is the tagset, that is, the set of all PoS tags.

In order to calculate emission probabilities, the number of times an ambiguity class is emitted by a given tag is approximated by means of:

$$\tilde{n}(\sigma, \gamma) \cong \sum_{s\in S}\sum_{g_i\in T(s)} C_{s,g_i}(\sigma, \gamma)\, p(g_i|s) \quad (5)$$

where $C_{s,g_i}(\sigma, \gamma)$ is the number of times ambiguity class σ is emitted by tag γ in path g_i of segment s. Therefore, the HMM parameter $b_{\gamma_i\sigma}$ corresponding to the emission probability of ambiguity class σ from the state associated with γ_i is computed as:

$$b_{\gamma_i\sigma} = \frac{\tilde{n}(\sigma, \gamma_i)}{\sum_{\sigma':\gamma_i\in\sigma'}\tilde{n}(\sigma', \gamma_i)} \quad (6)$$

Notice that when training from tagged texts the expressions used to compute transition and emission probabilities are analogous to previous equations, but in this case $p(g_i|s) = 1$ in (3) and (5) as only one path is possible in a tagged corpus segment; therefore, (3) and (5) would not be approximate anymore, but exact.

SL text segmentation must be carefully designed so that two words which get joint treatment in some stage of processing of the MT system are not associated to different segments. This would result in incorrect sequences in TL (for example, if two words involved in a word reordering rule are assigned to different segments) and, as a consequence of that, in wrong likelihood estimations. In general, HMMs can be trained by breaking the corpus into segments whose first and last word are unambiguous, since unambiguous words reveal or *unhide* the hidden state of the HMM (Cutting et al. 1992, sect. 3.4). Adequate strategies for ensuring segment independence depend on the particular translation system (we will describe in section 4 the strategy used in our experiments).

3.2 Cooperative learning of HMM

As mentioned in the introduction, we have already studied the use of a classical trigram model of words in TL texts as TL model. This implied that complete translation into the TL had to be performed for each possible disambiguation of each segment. One of the main obstacles encountered in following this approach is the inability of a word trigram model to distinguish *free rides* (words producing the same translation in TL for two or more disambiguation choices).

In this paper, however, we propose a completely different approach which reduces the negative effects of free rides because it uses a TL model of tags instead of words. As a consequence of that, morphological generation is skipped when performing the translation.

Consider the situation where one wants to compute the parameters for the two PoS taggers in a bidirectional machine translation (MT) system translating between languages A and B. The approach presented in the beginning of section 3 can be used for training the HMM corresponding to the PoS tagger for A by using some language model for B. The idea here is to use as such a model the HMM corresponding to the PoS tagger for B. The process is iteratively repeated by interchanging the roles of A and B. Therefore, both taggers learn cooperatively and are refined after each iteration by mutual interaction.

The TL model is thus a model of PoS tags in the current TL. The paths resulting for each possible disambiguation in a SL segment are *translated* into the corresponding (unambiguous) PoS tag sequence in TL, whose likelihood is estimated by computing the probability of the only possible state sequence in the HMM (see figure 1 in section 3.1) using only the transition probabilities $a_{\gamma_i\gamma_j}$.

Let $M_A[k]$ (resp. $M_B[k]$) be the HMM for language A (resp. B) computed at iteration k. At every iteration, the model $M_A[k]$ is trained by using statistical information from $M_B[k-1]$; after that, $M_B[k]$ is trained by considering the brand-new model $M_A[k]$ as a model for TL. The whole process can be graphically explained as follows:

$$M_B[0] \to M_A[1] \to M_B[1] \to M_A[2] \to \cdots \to M_B[k-1] \to M_A[k] \to M_B[k] \to \cdots$$

In principle, one would expect the initialization for $M_B[0]$ to be crucial. In order to determine its effect, two different models will be studied in the experiments: on the one hand, a "good" initial model obtained through the classical Baum-Welch algorithm; on the other hand, a "bad" initial model resulting from equiprobabilities.

The whole process runs in an unsupervised manner needing only two untagged corpora, one in A and one in B, not necessarily mutual translation.

4 Experiments

4.1 Machine Translation Engine

Since our training algorithm assumes the existence of a MT system (most likely, the system in which the resulting PoS tagger will be embedded), we briefly introduce the system used in the experiments, although almost any other architecture (with a HMM-based PoS tagger and a separate morphological generation phase) may also be suitable for the algorithm.

We used the publicly accessible Spanish–Catalan (two related languages) MT system interNOSTRUM (Canals-Marote et al. 2001), which basically follows a (morphological) transfer architecture consisting of the following sequence of stages:

- A *morphological analyser* 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* (categorial disambiguator) chooses, using a hidden Markov model (HMM), one of the LFs corresponding to an ambiguous SF.

- A *lexical transfer* module reads each SL LF and delivers the corresponding TL LF.
- A *structural transfer* module (parallel to the lexical transfer) uses a finite-state chunker to detect patterns of LFs which need to be processed for word reorderings, agreement, etc.
- A *morphological generator* delivers a TL SF for each TL LF, by suitably inflecting it, and performs other orthographical transformations such as contractions.

4.2 Text Segmentation

An adequate strategy for SL text segmentation is necessary. Besides the general rules mentioned in section 3, in our setup it must be ensured that all words in every pattern transformed by the structural transfer belong to the same segment.

The strategy followed in this paper is that of segmenting at unambiguous words, but keeping in the same segment those words that could get joint treatment by the structural transfer.

4.3 Results

We study PoS tagging performance for the Spanish (language B) and Catalan (language A) taggers after using the method described in previous sections. The process of iterative cooperative training is repeated until the PoS error rate corresponding to neither language improves in a series of 2 iterations.

The tagset used by the Spanish lexical disambiguator consists of 82 coarse tags (69 single-word and 13 multiword tags for contractions, verbs with clitic pronouns, etc.) grouping the 1 917 fine tags (386 single-word and 1 531 multiword tags) generated by the morphological analyser. The number of observed ambiguity classes is 249. The figures for Catalan are: 78 coarse tags (65 single-word and 13 multiword tags) grouping 1 977 fine tags (446 single-word and 1 531 multiword tags), and yielding 258 ambiguity classes. In addition, a few very frequent ambiguous words are assigned special hidden states (Pla & Molina 2004), and consequently special ambiguity classes.

As already commented, we also study whether the initial model $M_B[0]$ affects final performance. We consider two different initial models:

- A *good* HMM trained from untagged SL corpora following the common approach, that is, initializing the parameters of the HMM by means of Kupiec's method (Kupiec 1992) and using the Baum-Welch algorithm to reestimate the model; a 1 000 000-word ambiguous corpus was used for training. The resulting PoS tagger was tested after each iteration and the one giving an error rate which did not improved in the subsequent 3 iterations was chosen for evaluation; proceeding in this way, we prevent the algorithm from stopping if a better PoS tagger can still be obtained. The PoS error rate for this tagger when operating as a stand-alone tagger is 34.2%.
- A *bad* HMM with equiprobable transition and emission probabilities; this is the statistically-sound model having the least information. The PoS error rate for this tagger when operating as a stand-alone tagger is 76.5%.

For comparison purposes, we consider the performance in isolation of a Baum-Welch-trained HMM for each corpus, as mentioned in the preceding paragraph. Moreover, another HMM was trained from a tagged 20 000-word SL corpus and used as a reference for the best attainable results.

Experiments have been performed for different corpus sizes in order to test the amount of text required by the method. In addition, for each corpus size 3 different independent corpora are considered.

The PoS tagging error is evaluated in all cases using independent tagged corpora having $8\,031$ words in the case of Spanish and $8\,766$ in the case of Catalan. The percentage of ambiguous words (according to the lexicon) in the Spanish test corpus is 26.71% and the percentage of unknown words is 1.95% (29.39% and 2.42%, respectively, in the case of Catalan). For translation evaluation, an $8\,035$-word Spanish corpus, an $8\,863$-word Catalan corpus, and the corresponding human-corrected translations into the other language are used.

Table 1 shows the size of the training corpus, the average PoS tagging error rate and the average number of iterations necessary for the convergence of the new method when initializing $M_B[0]$ with the (good) Baum-Welch algorithm as described above. PoS tagging errors are expressed as the percentage of incorrect tags assigned to *ambiguous words* (including unknown words), not as the overall (over ambiguous and unambiguous words) percentage of correct tags; translation errors, however, do not consider unknown words and are expressed as the percentage of words (over ambiguous words) that need to be corrected or inserted when post-editing the translation because of wrongly tagged words.

Compare the results in table 1 to the PoS error rates attained with the standard (unsupervised) Baum-Welch-trained HMM ($31.7\%\pm3.4$ for Spanish and $37.8\%\pm0.7$ for Catalan) and with a HMM trained from unambiguous (supervised) texts (10.3% for Spanish and a slightly higher[1] percentage for Catalan). The translation error obtained with the former is $8.4\% \pm 0.8$ for Spanish and $13.6\% \pm 0.1$ for Catalan; the translation error of the latter is 2.6% for Spanish and a slightly higher percentage for Catalan. Our method's error lies between both models but it is still unsupervised.

Surprisingly, the results obtained when using a (bad) HMM with equiprobable parameters as the initial model $M_B[0]$ (probably, the worst possible initialization) are quite similar to those shown in table 1, although a few (usually around a couple of them) additional iterations are needed. For example, when considering the $5\,000$-word corpora, the PoS error rate is $28.2\% \pm 1.0$ for Spanish and $30.8\%\pm1.3$ for Catalan, whereas the translation error is $6.6\%\pm1.1$ for Spanish and $7.6\%\pm0.9$ for Catalan. Our algorithm is not too sensitive to the particular initial parameters used at the start, being able to learn by itself basically with no external initial help.

Concerning execution time, the new method needs higher training time than the Baum-Welch algorithm because of the enormous number of path likelihoods that need to be explicitely considered (remember, however, that the time necessary for processing ambiguous texts after training is independent of the algorithm being used to estimate the parameters of the HMM). For example, the overall training time in the case of corpora with $5\,000$ words is around one hour whereas the Baum-Welch algorithm usually takes a few minutes. The number of paths n_p and, consequently, the number of segment translations grows exponentially with segment length l as expected and can

[1]The authors do not have enough tagged text for Catalan, so reliable accurate results can not be given.

Corpus size (words)	Average PoS error		Average translation error		Average iterations
	Spanish	Catalan	Spanish	Catalan	
1 000	$29.0\% \pm 2.5$	$34.8\% \pm 1.0$	$8.9\% \pm 0.9$	$10.3\% \pm 0.3$	3.3
2 500	$27.1\% \pm 1.5$	$32.1\% \pm 0.7$	$6.6\% \pm 0.9$	$8.8\% \pm 0.1$	5.3
5 000	$27.4\% \pm 2.2$	$31.6\% \pm 1.4$	$6.6\% \pm 1.0$	$7.2\% \pm 0.4$	2.3
10 000	$27.8\% \pm 1.3$	$30.8\% \pm 1.6$	$6.2\% \pm 1.1$	$7.3\% \pm 0.5$	2.0
15 000	$27.8\% \pm 1.1$	$30.4\% \pm 1.6$	$6.0\% \pm 1.0$	$6.8\% \pm 0.3$	2.0
20 000	$28.2\% \pm 0.5$	$30.2\% \pm 0.8$	$5.8\% \pm 0.9$	$6.6\% \pm 0.1$	2.0
25 000	$27.9\% \pm 0.7$	$30.0\% \pm 0.8$	$6.0\% \pm 0.8$	$6.5\% \pm 0.2$	2.0
30 000	$29.9\% \pm 0.8$	$30.2\% \pm 0.8$	$6.0\% \pm 0.7$	$6.6\% \pm 0.2$	2.3

Table 1: PoS error rate, translation error rate and number of iterations (k in the text) needed for obtaining the two final taggers. PoS error rates are expressed as the percentage of incorrect tags assigned to *ambiguous words* (including unknown words). Translation errors are expressed as the percentage of words that need to be corrected (post-edited) due to mistaggings. The average PoS error rate with a HMM trained with the Baum-Welch algorithm is $31.7\% \pm 3.4$ for Spanish and $37.8\% \pm 0.7$ for Catalan; the average translation error in this case is $8.4\% \pm 0.8$ for Spanish and $13.6\% \pm 0.1$ for Catalan.

be approximated (for Spanish) by $n_p \approx 1.46^l$, where 1.46 can be interpreted as the average number of tags received by words.

5 Discussion

The most important conclusion is that unsupervised training of HMM-based PoS taggers using information from TL texts is feasible and opens a new line of research in the construction of PoS taggers. In particular, we show that a self-sufficient unsupervised training method can be designed for bidirectional MT systems, such that it fully bootstraps from scratch and ends up producing a reasonable PoS tagger for each of the two involved languages. Although the results presented here are for two similar languages, we expect them to hold for more dissimilar language pairs, since most translation divergences would then be handled by modules after de PoS tagger.

The proposed method needs a relatively small number of words (as seen in the last section, 5 000 words on each side is enough) if compared with common corpus sizes used in the Baum-Welch algorithm. In any case, both methods run in an unsupervised manner, avoiding the need for tagged corpora. As stated, tagging errors lie between those produced by classical SL-based unsupervised models using Baum-Welch estimation and those attained with a supervised solution based on unambiguous texts.

The training method described in this paper produces a PoS tagger which is in tune not only with SL but also with the TL of the translation engine. The method may also be used to customize the MT engine to a particular text type or subject or to statistically "retune" it after introducing new transfer rules in the MT system.

6 Future Work

We plan to research on better estimates for $p(g_i|\tau(g_i, s))$ in (1), even though with the TL models used here it is quite unlikely that $\tau(g_i, s) = \tau(g_j, s)$ for $g_i \neq g_j$. As already mentioned, equation (2) is a very simple approximation, but more elaborated expressions may worth studying and improve performance.

A different line of work will focus on time complexity reduction. We propose to use a k-best Viterbi algorithm for the current model in order to calculate approximate likelihoods so that only the most promising disambiguation paths are considered, thus reducing the number of translations to be computed.

Acknowledgments

Work funded by the Spanish Government through grants TIC2003-08681-C02-01 and BES-2004-4711. We thank Rafael C. Carrasco for very useful comments about the method presented here.

References

Canals-Marote, R. et al.: 2001, 'The Spanish-Catalan machine translation system interNOSTRUM', in *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, pp. 73–76.

Cutting, D., J. Kupiec, J. Pedersen & P. Sibun: 1992, 'A practical part-of-speech tagger.' in *Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference.*, Trento, Italy, pp. 133–140.

Hutchins, W.J. & H.L. Somers: 1992, *An Introduction to Machine Translation*, London: Academic Press.

Kupiec, J.: 1992, 'Robust part-of-speech tagging using a hidden Markov model', *Computer Speech and Language*, **6**(3): 225–242.

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

Rabiner, L. R.: 1989, 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE*, **77**(2): 257–286.

Sánchez-Martínez, F., J.A. Pérez-Ortiz & M.L. Forcada: 2004, 'Exploring the use of target-language information to train the part-of-speech tagger of machine translation systems', *Proceedings of EsTAL, España for Natural Language Processing*, Alicante, Spain, accepted.

Yarowsky, D. & G. Ngai: 2001, 'Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection Across Aligned Corpora', in *Proceedings of NAACL-2001*, Pittsburgh, PA, USA, pp. 200–207.