Target-Language-Driven Agglomerative Part-of-Speech Tag Clustering for Machine Translation*

Felipe Sánchez-Martínez and **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

This paper presents a method for reducing the set of different tags to be considered by a part-of-speech tagger. The method is based on a clustering algorithm performed over the states of a hidden Markov model, which is initially trained by considering information not only from the source language, but also from the target language, using a new unsupervised technique which has been recently proposed to obtain taggers involved in machine translation systems. Then, a bottom-up agglomerative clustering algorithm groups the states of the hidden Markov model according to a similarity measure based on their transition probabilities; this reduces the complexity by grouping the initial finer tags into coarser ones. The experiments show that part-of-speech taggers using the coarser tags have smaller error rates than those using the initial finest tags; moreover, considering unsupervised information from the target language results in better clusters compared to those unsupervisedly built from source language information only.

1 Introduction

This paper explores the automatic induction of hidden Markov model (HMM) topologies used for part-of-speech tagging in a machine translation (MT) system. Hidden Markov models (Rabiner 89) have been widely used for part-of-speech (PoS) tagging (Cutting *et al.* 92). In this case, the HMM topology is usually fixed (that is, manually defined following linguistics guidelines) and the training phase is restricted to the estimation of probabilities.

There have been some attempts to define the HMM topology automatically. (Stolcke & Omohundro 94) describe a technique for inducing the HMM structure from data, which is based in the general *model merging* strategy (Omohundro 92), but their work focuses on HMMs for speech recognition, not on HMMs used for PoS tagging where some additional restrictions have to be taken into account. On the other hand, the model merging method starts with a maximum likelihood HMM that directly encodes the training data, that is, where there is exactly one path for each element in the training corpus, and each path is used by one element only. This approximation is not a feasible approach when the resulting HMM will be used in a real environment such as a MT system, in which previously unseen events might occur.

A later work (Brants 95) focuses on the problem of finding the structure of a HMM used for PoS tagging. In that work the author also follows the model merging technique to find the tagset (set of PoS tags) to be used, but this time taking into account some restrictions in order to preserve the information provided by the fine states the initial HMM has. Furthermore, in this work the initial model has one state per part-of-speech, not per word occurrence, but it is trained following a supervised method.

In this paper we explore the use of a bottom-up agglomerative clustering algorithm to obtain the tagset to be used in a HMM-based PoS tagger within a MT system. The initial model is the one obtained using the fine tags delivered by the morphological analyzer of the MT system, trained following an unsupervised method that takes into account information from the target language (TL) (Sánchez-Martínez *et al.* 04a; Sánchez-Martínez *et al.* 04b) to estimate the HMM parameters. We apply the agglomerative clustering procedure both to taggers trained using the TL-driven procedure above and to taggers unsupervisedly trained using the Baum-Welch (Baum 72) algorithm.

The paper is organized as follows: Section 2 overviews the use of HMM for part-of-speech (PoS) tagging. In section 3 the principles of the TL-driven HMM training method are explained; then, in section 4 the clustering strategy is described, and section 5 explains the shallow-transfer MT system used for the TL-driven train-

* 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 grant BES-2004-4711.

ing method and the experiments conducted. Finally, in sections 6 and 7 the results are discussed and future work is outlined.

2 Hidden Markov models for part-of-speech tagging

In this section we overview the application of HMMs in the natural language processing field as PoS taggers.

A HMM (Rabiner 89) is defined as $\lambda = (\Gamma, \Sigma, A, B, \pi)$, where Γ is the set of states, Σ 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 σ being emitted from each state γ, and the vector π, 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 HMM is used to perform PoS tagging, each HMM state γ is made to correspond to a different PoS tag,[1] and the set of observable outputs Σ are made to correspond to *word classes*. Typically a word class is an *ambiguity class* (Cutting *et al.* 92), 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 of the initial probability of each state can be avoided by assuming that each sentence begins with the end-of-sentence mark. In this case, $\pi(\gamma)$ is 1 when γ 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 (Cutting *et al.* 92) and (Manning & Schütze 99, ch. 9).

3 Target-language training overview

Typically the training of HMM-based PoS taggers is done using the *maximum-likelihood estimate* (MLE) (Gale & Church 90) method when tagged corpora[2] are available (supervised method) or using the Baum-Welch algorithm with untagged corpora[3] (unsupervised method). But, when the resulting PoS tagger is to be embedded as a module of a working MT system, the HMM training can be done in an unsupervised way using information not only from the source-language (SL), but also from the TL. This new training method has been previously described in (Sánchez-Martínez *et al.* 04a; Sánchez-Martínez *et al.* 04b), and is the method used to obtain the initial model that uses the largest possible tagset (that is, the one using the finest possible tags).

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 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 TL statistical 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 model being trained.

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 choices or PoS combinations, that is to say:

- $\boldsymbol{g_1}$ = (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).

The next step is to translate the SL segment into the TL according to each disambiguation $\boldsymbol{g_i}$:

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

[1] This is only true when a first-order HMM is considered. In an n-th order HMM each state corresponds to a sequence of n PoS tags.

[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 a morphological analyzer) the set of all possible PoS tags independently of context.

- $\tau(\boldsymbol{g_4}, s) =$"*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. Finally, the calculated probabilities for each disambiguation $\boldsymbol{g_i}$ are used to estimate the HMM parameters through the MLE method as if they were fractional counts.

4 Tagset clustering strategy

The reason for reducing the number of tags used by PoS taggers is due to the fact that the less tags the tagset has the better the HMM parameters are estimated, through the reduction of the data sparseness problem. Furthermore, as the number of transition probabilities to estimate is, for a first order HMM, quadratic with the number of tags, the number of parameters to store may be drastically reduced.

In order to obtain a coarser tagset we have not followed the model merging strategy already used by Brants (Brants 95) because it is a very time consuming method. Instead, we perform a bottom-up agglomerative clustering on an initial HMM that has as many states as different fine PoS tags the morphological analyzer delivers (see section 5 for details about the different PoS tags delivered by the morphological analyzer).

Bottom-up agglomerative clustering has been used for HMM state clustering (Rivlin *et al.* 97) in speech recognition tasks. One advantage of this clustering algorithm is that the number of clusters (coarse tags) to discover is automatically determined by providing the algorithm with a distance threshold. The algorithm begins with as many clusters as fine tags there are, and in each step those clusters that are closer are merged into a single one only if an additional constraint (see below) is met. The clustering stops when there are no clusters to be merged because their distance is larger than the specified threshold, or the constraint does not hold.

4.1 Constraint on the clustering

A very important property of the resulting tagset is that it must be possible to restore the original information (all grammatical features) represented by the fine tag from the coarser one; note that this is the information we are interested in, as it is used by the subsequent MT modules to carry out the translation. To ensure this property a constraint must hold; this constraint, already used in (Brants 95), establishes that two tags (states) cannot be merged in the same cluster if they share the emission of one or more word class (observables) outputs. This is because in this case, the PoS tagger would not be able to decide on a PoS tag for the observable output.

The previous constraint can be formally described as follows. Let f be a fine tag, c a coarse tag (cluster), σ an observable output, and F, C and Σ the fine tagset, the coarse one and the set of observable outputs, respectively. The original information of the fine tag f can be retrieved from the coarse one c by means of the injective function h defined as:

$$h : \Sigma \times C \rightarrow F \tag{1}$$

To ensure that this function is injective, that is, that for a given observable σ and a given coarse tag c there is only one fine tag f, the next constraint must be met:

$$\forall c \in C, \sigma \in \Sigma, f_1, f_2 \in c, f_1 \neq f_2 : f_1 \in \sigma \Rightarrow f_2 \notin \sigma, \tag{2}$$

where with $f \in c$ we mean that the fine tag f is in the cluster denoted by c, and with $f \in \sigma$ we mean that the observable output σ can be emitted from the fine tag f.

If the constraint expressed in (2) holds, function h is injective, and no information is lost when grouping fine tags into coarser ones.

4.2 Distance between clusters

As an agglomerative clustering will be applied, a distance measure between two clusters is needed in order to measure how similar they are.

Before defining how the distance between two clusters is calculated, let us define how the distance between two fine tags is calculated. The distance between two fine tags is based on the Kullback-Leibler *directed logarithmic divergence* (Kullback & Leibler 51) applied to the probabilistic distributions defined by the transition probabilities A between each fine tag and the rest. The directed logarithmic divergence measures the relative entropy between two probabilistic distributions $p(x)$ and $q(x)$:

$$d(p, q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)} \tag{3}$$

Since $d(p,q) \neq d(q,p)$, the relative entropy is not a true metric, but it satisfies some important mathematical properties: it is always nonnegative and equals zero only if $\forall x\, p(x) = q(x)$.

As for the clustering algorithm a symmetric distance measure is needed, we use the *intrinsic discrepancy* (Bernardo & Rueda 02) defined as:

$$\delta(p,q) = \min(d(p,q), d(q,p)) \tag{4}$$

Another possibility to make the distance measure symmetric would be to use the *divergence* (Brants 96) defined as

$$\mathrm{Div}(p,q) = d(p,q) + d(q,p) \tag{5}$$

but the intrinsic discrepancy is preferred, among other reasons, because if one probabilistic distribution has null values for some range of X and the other has not, the intrinsic discrepancy is still finite while the divergence is not.

Now that we know how to calculate the distance between two fine tags, we define the way in which the distance between two clusters is calculated. As the intrinsic discrepancy used does not hold the *triangle inequality*, the search space is not a metric one, and calculating a representative for each cluster is not a trivial task. Because of this, the distance between two clusters will be the *unweighted pair-group average*:

$$\delta(c_1,c_2) = \frac{\sum_{t_1 \in c_i} \sum_{t_2 \in c_2} \delta(t_1,t_2)}{\mathrm{card}(c_1)\mathrm{card}(c_2)}, \tag{6}$$

although other distances such as the *weighted pair-group average* or the *minimum/maximum pair-group distance* could also be suitable.

5 Experiments

As has been already mentioned, before applying the clustering algorithm a HMM-based PoS tagger for Spanish is trained using the fine tags delivered by the morphological analyzer. These fine tags have all the morphological information used by the rest of the modules of the MT system. For example, the Spanish word *señal* has the next morphological analysis (fine tag): "noun, feminine, singular", which is different from the fine tag "noun, feminine, plural" given for the word *señales*.

As the previous example illustrates, fine tags discriminate gender, number or, in a verb case, the person who performs the action, among other grammatical features. This causes the number of fine tags to be very large: 1 328 fine tags grouped into 1 594 ambiguity classes in our Spanish lexicon. Notice that the number of HMM transition probabilities to be estimated is quadratic with the number of tags, and the larger the tagset the worse the data sparseness problem.

We have conducted two different experiments for Spanish, one with the initial model trained using information from the TL,[4] as already explained above, and another one in which the initial model is trained using the classical Baum-Welch algorithm; in both cases the training is fully unsupervised.

As has been mentioned, in order to train a HMM-based PoS tagger using information from the TL a working MT system is required. In the next section we overview the MT system used in our experiments. Then we report the results achieved by the TL-driven training method and the Baum-Welch algorithm with the fine tagset, and the results achieved with the tagsets automatically obtained through the bottom-up agglomerative clustering already discussed.

5.1 Machine translation engine

Now we briefly introduce the MT system used in the experiments, although almost any other MT architecture (using a HMM-based PoS tagger) may also be suitable for the TL-driven training algorithm.

We used the Spanish–Catalan (two related languages) MT system interNOSTRUM[5] (Canals *et al.* 00) which basically follows a shallow transfer architecture consisting of the following sequence of stages:[6]

- A *morphological analyzer* 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. The lexical category and the morphological inflection information constitute the fine tag for each LF.
- A *PoS tagger* chooses, using a hidden Markov

[4] For the experiments we use as a TL model a classical trigram language model like the one used in (Sánchez-Martínez *et al.* 04b)

[5] The MT system and the morphological analyzer may be accessed at `http://www.internostrum.com`.

[6] A complete rewriting of this MT engine (Corbí-Bellot *et al.* 05) has been recently released under an open source license (`http://apertium.sourceforge.net`).

Training method	Avg. PoS error
Baum-Welch	$28.7 \pm 2.0\%$
TL based	$25.5 \pm 0.3\%$

Table 1: Average PoS tagging error rate (over ambiguous words only, and without considering unknown words) for the initial HMM that uses the large fine tagset. The error rate reported when the Baum-Welch training algorithm is used is the result of the best of 100 iterations. As can be seen, the standard deviation for the Baum-Welch algorithm is much larger than for the TL-driven algorithm, this is because the Baum-Welch algorithm can fall in a local maxima for some corpora.

model (HMM), one of the LFs corresponding to an ambiguous SF. This is the module whose training is considered in this paper.

- 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. and performs these operations.
- A *morphological generator* 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 applied the presented bottom-up agglomerative clustering on a HMM previously trained using the large (indeed largest possible) initial tagset. Once the initial HMM has been trained the transition probabilities A are used to obtain the coarser tagset. Note that the final number of coarse tags is indirectly determined because the clustering algorithm is provided with a distance threshold.

The experiments have been done with three different corpora in order to know how the clustering algorithm behaves. When using the Baum-Welch algorithm to train the initial model we use three disjoint corpora with around 1 000 000 words each. For the TL-driven training method the corpora used were smaller, around 300 000 words each, because the training algorithm takes much more time, and convergence was reached before processing the whole 300 000 words.

Table 1 shows the average PoS tagging error rate for the two training methods used to obtain the initial HMM used to perform the bottom-up agglomerative clustering. As may be seen, the results achieved by the TL-driven training method are (expectedly) better as was already reported in previous works (Sánchez-Martínez *et al.* 04b). The error rates reported in Table 1 are over ambiguous words only, not over all words, and do not take into account unknown words. The PoS tagging error rate is evaluated using an independent 8 031-word hand-tagged Spanish corpus. The percentage of ambiguous words in that corpus is 26.7% and the percentage of unknown words is 2.0%.

In order to find the threshold that produces the best tagset we have performed the bottom-up agglomerative clustering for thresholds varying from 0 to 2.5 in increments of 0.05. Figure 1 shows the evolution of the PoS tagging error rate with the threshold for one of the corpora used (the remaining two corpora behave in a similar way, the error rate improvement being slightly lower) when using the TL-driven training method to obtain the initial HMM. The PoS tagging error corresponding to the negative threshold is the error rate of the initial HMM using the largest tagset. In that figure the number of coarse tags obtained automatically with each threshold is also shown. It has to be noted that after applying the clustering algorithm the HMM parameters are recalculated using the fractional counts collected during the TL-driven training (this would be equivalent to retraining with the new tagset). Thus, there is no need to retrain the model for each tagset; one simply recalculates the transition and emission probabilities.

As can be seen in Figure 1, with a null threshold value the number of clusters is 327, that is, there are around 1 000 fine tags that have exactly the same transition probabilities. This is because these fine tags are mostly for verbs receiving one (*dame* = "give+me") or two (*dámelo* = "give+me+it") enclitic pronouns, which rarely appear in the training corpus; therefore, the clustering algorithm puts all these fine tags in the same cluster. Furthermore, it can be seen that the best PoS tagger is obtained with a threshold of 1.25, which produces a tagset with only 241 coarse tags. The 241-tag tagset groups in the same cluster, for example, the third person singular tonic pronouns (*consigo* = "with himself/herself/itself", *usted* = "you"), the third person masculine plural tonic pronoun (*ellos* =

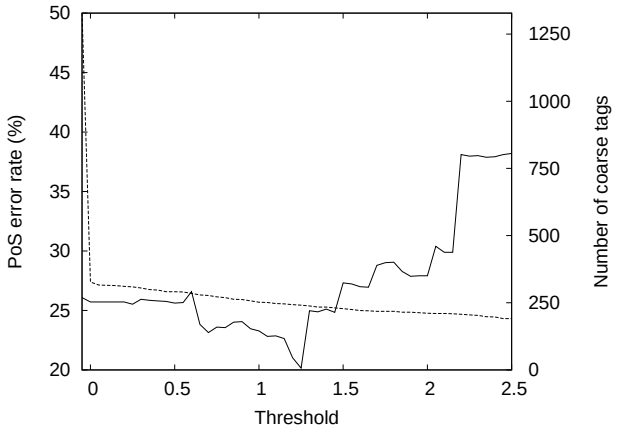

Figure 1: Evolution of the PoS tagging error (solid line with values on the left vertical axe) according to the different threshold values for $d(c_1, c_2)$ used in the experiments, when using as an initial model the one obtained with the TL-driven training method. The number of tags of the obtained tagset for each threshold is also given (dotted line with values on the right vertical axe).

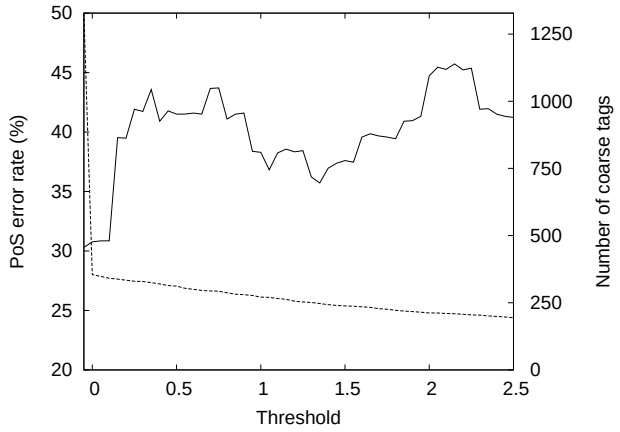

Figure 2: Evolution of the PoS tagging error (solid line with values on the left vertical axe) according to the different threshold values for $d(c_1, c_2)$ used in the experiments, when using as an initial model the one obtain with the Baum-Welch algorithm. The number of tags of the inferred tagset for each threshold is also given (dotted line with values on the right vertical axe).

"they"), the third person neutral tonic pronoun (*ello* = "it"), the third singular tonic pronouns (*nadie* = "no one", *alguien* = "someone", etc.), the third person reflexive tonic pronoun (*sí* = "himself/herself/itself"), and the relative *quien* (= "who/whom"). Furthermore, contrary to what it may be expected some specializations of the same category (for example, feminine adjective and masculine adjective) are assigned to different coarse tags (clusters).

Figure 2 shows the evolution of the PoS tagging error rate and the number of tags for each of the inferred tagset when the initial model is the one obtained using the Baum-Welch algorithm on one of the corpora used (the other two corpora behave in the same way). In this case, after running the clustering algorithm, the HMM was retrained with the new tagset for 100 Baum-Welch iterations.[7] The PoS tagging error rate given in that figure for each threshold is the one provided by the best Baum-Welch iteration. Notice that because of the presence of local maxima in which the Baum-Welch algorithm can fall, the PoS tagging error rate may behave erratically.

As can be seen in Figure 2 clustering does not improve the PoS tagging error rate, and the number of tags of the obtained tagsets for the same threshold values is similar to the number of tags obtained from the TL-trained initial model.

[7]In principle, one could also recalculate the probabilities from the forward-backward auxiliary variables, but we found it easier to simply retrain.

6 Discussion

We have explored the automatic tagset reduction, starting from a large fine tagset, by means of a bottom-up agglomerative clustering algorithm. We have conducted two different experiments: one that uses the Baum-Welch algorithm to obtain the initial HMM with all the fine tags, and another one that uses information from the TL to obtain that initial model.

The results reported show that using the TL-driven training method slightly improves the tagging accuracy, proving that the TL-driven training method is a good unsupervised approach that gives better results than the classical Baum-Welch algorithm.

In the experiments reported in this paper we have not used any smoothing technique to avoid null transition and emission probabilities for those unseen events in the training corpus.

Preliminary experiments using the *expected-likelihood estimate* (ELE) method (Gale & Church 90), which use a very rudimentary smoothing technique, show that the resulting coarse tagset is smaller for equal threshold values. We plan to test whether this still happens when applying a smoothing technique in the maximization step of the Baum-Welch algorithm.

7 Future work

The bottom-up agglomerative clustering uses a distance between clusters. In this paper we have

used the unweighted pair-group average of the intrinsic discrepancy, but other distance measures could also be suitable. We plan to test the minimum pair-group distance which is reported to produce clusters with more disperse elements and the maximum pair-group distance which usually gives more compacted clusters.

In this paper the *intrinsic discrepancy* was used to measure the distance between two fine tags. This measure is finite if one distribution has null values in some range of X and the other not. But, when one probabilistic distribution has null values where the other does this measure becomes infinity. In order to avoid this problem we plan to use the Jensen-Shannon divergence (Grosse *et al.* 02) which is finite for all pairs of distributions.

In one of the papers presenting the TL-driven training method (Sánchez-Martínez *et al.* 04b) the coarse tagset used was manually defined following linguistic guidelines and the method behaved unstably because of the *free-ride phenomenon* (different disambiguations leading to the same translation). We plan to test whether this problem persists with the best automatically inferred tagset.

References

(Baum 72) L. E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

(Bernardo & Rueda 02) J. M. Bernardo and R. Rueda. Bayesian hypothesis testing: A reference approach. *International Statistical Review*, 70:351–372, 2002.

(Brants 95) Thorsten Brants. Tagset reduction without information loss. In *33rd Annual Meeting of the Association for Computational Linguistics (ACLANNUAL'95)*, Cambridge, Massachussetts, USA, 1995.

(Brants 96) Thorsten Brants. Estimating Markov model structures. In H. T. Bunnell and W. Idsardi, editors, *4th International Conference on Spoken Language Processing (ICSLP'96), October 3-6*, volume 2, pages 893–896, Philadelphia, USA, 1996.

(Canals *et al.* 00) Raül Canals, Anna Esteve, Alicia Garrido, M. Isabel Guardiola, Amaia Iturraspe-Bellver, Sandra Montserrat, Pedro Pérez-Antón, Sergio Ortiz, Herminia Pastor, and Mikel L. Forcada. InterNOSTRUM: a Spanish-Catalan machine translation system. *Machine Translation Review*, 11:21–25, 2000.

(Corbí-Bellot *et al.* 05) Antonio M. Corbí-Bellot, Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, and Kepa Sarasola. 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*, pages 79–86, Budapest, Hungary, May 2005.

(Cutting *et al.* 92) D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference.*, pages 133–140, Trento, Italia, 1992.

(Gale & Church 90) William A. Gale and Kenneth W. Church. Poor estimates of context are worse than none. In *Proceedings of a workshop on Speech and natural language*, pages 283–287. Morgan Kaufmann Publishers Inc., 1990.

(Grosse *et al.* 02) Ivo Grosse, Pedro Bernaola-Galván, Pedro Carpena, Ramón Román-Roldán, Jose Oliver, and H. Eugene Standley. Analisys of symbolic sequences using the jensen-shannon divergence. *Physical Review E*, 65(4), 2002.

(Kullback & Leibler 51) S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Math. Stats.*, 22:79–86, 1951.

(Manning & Schütze 99) Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT press, 1999.

(Omohundro 92) Stephen M. Omohundro. Best-first model merging for dynamic learning and recognition. In John E. Moody, Steve J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 958–965. Morgan Kaufmann Publishers, Inc., 1992.

(Rabiner 89) L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

(Rivlin *et al.* 97) Ze'ev Rivlin, Ananth Sankar, and Harry Bratt. HMM state clustering across allophone class boundaries. In *Proc. Eurospeech '97*, pages 127–130, Rhodes, Greece, 1997.

(Sánchez-Martínez *et al.* 04a) F. Sánchez-Martínez, J. A. Pérez-Ortiz, and M. L. Forcada. Cooperative unsupervised training of the part-of-speech taggers in a bidirectional machine translation system. In *Proceedings of TMI, The Tenth Conference on Theoretical and Methodological Issues in Machine Translation*, pages 135–144, October 2004.

(Sánchez-Martínez *et al.* 04b) F. Sánchez-Martínez, J. A. Pérez-Ortiz, and M. L. Forcada. 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*, pages 137–148. Springer-Verlag, October 2004.

(Stolcke & Omohundro 94) A. Stolcke and S. M. Omohundro. Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, 1947 Center Street, Berkeley, CA, 1994.