Mach Translat (2008) 22:29–66
DOI 10.1007/s10590-008-9044-3

Using target-language information to train part-of-speech taggers for machine translation

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

Received: 28 January 2008 / Accepted: 27 October 2008 / Published online: 25 November 2008

Abstract Although corpus-based approaches to machine translation (MT) are growing in interest, they are not applicable when the translation involves less-resourced language pairs for which there are no *parallel* corpora available; in those cases, the rule-based approach is the only applicable solution. Most rule-based MT systems make use of part-of-speech (PoS) taggers to solve the PoS ambiguities in the source-language texts to translate; those MT systems require accurate PoS taggers to produce reliable translations in the target language (TL). The standard statistical approach to PoS ambiguity resolution (or *tagging*) uses hidden Markov models (HMM) trained in a supervised way from hand-tagged corpora, an expensive resource not always available, or in an unsupervised way through the Baum-Welch expectation-maximization algorithm; both methods use information only from the language being tagged. However, when tagging is considered as an intermediate task for the translation procedure, that is, when the PoS tagger is to be embedded as a module within an MT system, information from the TL can be (unsupervisedly) used in the training phase to increase the translation quality of the whole MT system. This paper presents a method to train HMM-based PoS taggers to be used in MT; the new method uses not only information from the source language (SL), as general-purpose methods do, but also information from the TL and from the remaining modules of the MT system in which the PoS tagger is to be embedded. We find that the translation quality of the MT system embedding a PoS tagger trained in an unsupervised manner through

F. Sánchez-Martínez (✉) · J. A. Pérez-Ortiz · M. L. Forcada
Dept. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, 03071 Alacant, Spain
e-mail: fsanchez@dlsi.ua.es

J. A. Pérez-Ortiz
e-mail: japerez@dlsi.ua.es

M. L. Forcada
e-mail: mlf@dlsi.ua.es

this new method is clearly better than that of the same MT system embedding a PoS tagger trained through the Baum-Welch algorithm, and comparable to that obtained by embedding a PoS tagger trained in a supervised way from hand-tagged corpora.

Keywords Rule-based machine translation · Part-of-speech tagging · Hidden Markov models · Language modeling

1 Introduction

Nowadays, with the growing availability of machine-readable monolingual and *parallel* bilingual corpora, corpus-based approaches to machine translation (MT), such as statistical MT (Brown et al. 1993; Koehn 2008) or example-based MT (Nagao 1984; Carl and Way 2003), are growing in interest. However, these approaches usually require large amounts (in the order of tens of millions of words) of parallel corpora (Och 2005) to build general-purpose MT systems of a reasonable translation quality. Given that such data sets are not always available, or exist only in small amounts, rule-based MT (RBMT) systems are still being actively developed since the rule-based approach is the only realistic approach to build MT systems of a reasonable quality in such cases; this is actually the situation for most less-resourced language pairs, such as Occitan–Catalan, French–Catalan and English–Afrikaans. In addition, the kind of errors RBMT systems produce are more predictable, which makes it easier to correct their output for dissemination purposes, and to diagnose them during development.

RBMT systems heavily depend on linguistic knowledge such as morphological and bilingual dictionaries (containing lexical, syntactic and even semantic information), part-of-speech (PoS) disambiguation rules or manually disambiguated corpora, and a large set of transfer rules; therefore, the process of building an RBMT system entails a huge human effort for building the necessary linguistic resources (Arnold 2003).

The main goal of the work we present in this paper is to make easier the development of RBMT systems that make use of PoS taggers in their analysis phase by avoiding the need for human intervention in the process of building these PoS taggers, which need to be accurate in order to obtain good translation results. With this aim, we explore the use of the following information in order to train hidden Markov model (HMM)-based PoS taggers for MT:

– source language (SL) information, as usual;
– target language (TL) information; and
– information in the rest of modules of the MT system in which the resulting PoS tagger is to be embedded.

Figure 1 shows a general RBMT system using a PoS tagger in its analysis phase, and how the PoS tagger relates to the rest of the MT architecture.

PoS tagging is a well-known task and a common step in many RBMT systems. A PoS tagger is a program that attempts to assign the correct PoS tag or *lexical category* to all words of a given text; typically, by relying on the assumption that a word can be assigned a single PoS tag by looking at the PoS tags of neighbouring words.

Usually PoS tags are assigned to words by looking them up in a lexicon, or by using a morphological analyzer (Merialdo 1994). A large portion of the words found in a

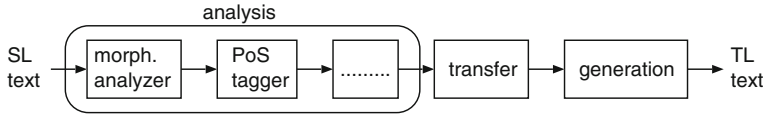

Fig. 1 A general RBMT system using a PoS tagger in its analysis phase

text have only one possible PoS tag, but there are *ambiguous* words that have more than one PoS tag;[1] for example, the word *book* can be either a noun (*She bought a book for you*) or a verb (*We need to book a room*). The choice of the correct PoS tag may be crucial when translating to another language because the translation of a word may greatly differ from one PoS to another; in the above example, the English word *book* may be translated into Spanish as *libro* or as *reservar* depending on the PoS (noun or verb, respectively).

1.1 Part-of-speech tagging approaches for machine translation

Different approaches have been followed in order to obtain robust general-purpose PoS taggers to be used in a wide variety of natural language processing (NLP) applications. On the one hand, rule-based approaches either learn automatically (Brill 1992, 1995b), or code manually, rules capable of solving the PoS ambiguity. On the other hand, statistical approaches (Dermatas and Kokkinakis 1995; Sánchez-Villamil et al. 2004) use corpora to estimate a probability model that is then used to perform the PoS tagging of new corpora.

The classical statistical approach to PoS tagging uses hidden Markov models (HMM) (Baum and Petrie 1966; Rabiner 1989; Cutting et al. 1992). These statistical models can be trained in a *supervised* way from *hand-tagged* (or simply *tagged*) corpora using *maximum-likelihood estimation* (MLE) (Gale and Church 1990). A tagged corpus is a text in which each PoS ambiguity has been solved by a human expert; therefore, tagged corpora are a very expensive linguistic resource which are not always available, especially for less-resourced languages.

If a tagged corpus is not available, HMMs can be trained in an *unsupervised* way by using *untagged* corpora as input to the Baum-Welch expectation-maximization (EM) algorithm (Baum 1972). An untagged corpus (Merialdo 1994) is a text in which each word has been assigned the set of all possible PoS tags that it could receive independently of the context. This kind of text can be automatically obtained if a morphological analyzer or a lexicon is available. In an untagged corpus, ambiguous words receive more than one PoS tag.

The two methods (supervised and unsupervised) mentioned above to train HMM-based PoS taggers only use information from the language being tagged, a natural approach when PoS tagging is to be applied in NLP applications involving only one language. However, when PoS taggers are used in MT, that is, when tagging is viewed

[1] In Romance language texts about one word out of three is usually ambiguous.

just as an intermediate task for the whole translation procedure, there are two points to which, as far as we know, the research community in general has not paid attention:

– on the one hand, that there is a natural source of knowledge, in addition to parallel corpora (Yarowsky and Ngai 2001; Dien and Kiem 2003), that can be used while training to obtain better PoS taggers; namely, the use of a statistical model of the TL; and
– on the other hand, that in MT PoS tagging is just an intermediate step needed to produce good translations into the TL; therefore, what really counts is translation quality rather than PoS tagging accuracy, i.e. one should not care whether a word is incorrectly tagged if it gets translated correctly.

1.2 Using target-language information to train part-of-speech taggers

This paper describes a new unsupervised training method aimed at producing PoS taggers to be used in MT. In order to train an HMM-based PoS tagger for the SL, besides using SL information, the method uses information from both the TL and from the remaining modules of the MT system in which the resulting PoS tagger is to be used; note, however, that these two additional sources of knowledge—the TL and the rest of the modules of the MT engine—are not used to tag the SL texts through the Viterbi algorithm (Rabiner 1989; Manning and Schütze 1999, p. 332) when translating. To our knowledge, the method—preliminary versions of which have been already presented in conference papers (Sánchez-Martínez et al. 2004a,b)— is the first one to use TL information to train an SL component in an MT system.

The main idea behind the use of TL information is that the correct disambiguation (tag assignment) of a given SL segment will produce a TL translation which is more likely than any (or most) of the translations produced from the remaining wrong disambiguations. As the resulting SL PoS tagger is intended to be used in MT, we focus on MT performance rather than on PoS tagging accuracy. In the experiments we compare the performance of our method with the classical unsupervised Baum-Welch EM algorithm; the results are better and the amount of SL text needed (for training) is smaller. In addition, for Spanish we compare the translation quality achieved by a PoS tagger trained with our method to that of a PoS tagger trained in a supervised way; surprisingly, translation quality is comparable for both methods. As a disadvantage, our method needs longer training times and an additional TL text corpus, which, however, does not need to be related (parallel or comparable) to the SL corpus.

In sum, the basic idea behind this method is to ease the development of an RBMT system by using, on the one hand, the linguistic information in some of its modules and, on the other hand, statistics about the TL to train its PoS tagger in an unsupervised way; therefore, we avoid the need to manually disambiguate the training corpus. Although the reader may think that this new method needs an MT system to exist, it is actually the other way round; developers building an RBMT system may use our method to unsupervisedly build the PoS tagger of that MT system. To that end, they only need to build the other modules of the translation engine before applying our method in order to obtain the PoS taggers to be used in that RBMT system.

1.2.1 Background

Yarowsky and Ngai (2001) proposed a method which also uses information from the TL in order to train PoS taggers. However, they considered information from *aligned* parallel corpora and from (at least) one manually tagged corpus for the TL. A similar approach is followed by Dien and Kiem (2003), who bootstrap a PoS-annotated English corpus via transformation-based learning (Brill 1995a) by exploiting the PoS information of the corresponding Vietnamese words in a Vietnamese–English parallel corpus. They then project the PoS annotations from the English side of the parallel corpus to the Vietnamese side through the word alignments. Finally they manually correct the resulting Vietnamese PoS-annotated corpus. In contrast, the method proposed in this paper needs neither aligned parallel corpora nor manually tagged texts. Moreover, our method views PoS tagging as an intermediate task for the translation procedure, instead of as an objective in its own right.

Foster et al. (1997) move the focus from the meaning of the SL text to the production of the corresponding TL text in an interactive MT system. Although their work is not directly related to the approach we present in this paper, it shares with it the idea that we should not be concerned about the SL meaning (in our case, the correct disambiguation of the SL text), but instead about the correctness and fluency of the translation.

Carbonell et al. (2006) proposed a new MT framework in which a large full-form bilingual dictionary and a huge TL corpus is used to carry out the translation; neither parallel corpora nor transfer rules are needed. The ideas behind the paper of Carbonell et al. and ours share the same principle; if the goal is to obtain good translations in the TL, let the TL decide whether a given "construction" in the TL is good or not. In contrast, the method of Carbonell et al. uses TL information at translation time, while ours only uses TL information when training one module that is then used, in conjunction with the rest of MT modules, to carry out the translation; therefore, no TL information is used by our method at translation time, which makes the whole MT system much faster.

1.2.2 Overview of the method

Our method works as follows:

- For a given segment (word sequence) s in the SL, all possible disambiguation choices (combinations of the PoS tags for each word) $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N$ are considered;[2]
- the SL segment s is translated into the TL according to each possible disambiguation $\boldsymbol{g}$ by using the modules of the MT system which follow the PoS tagger (see Fig. 1);
- each of the resulting translations $\tau(\boldsymbol{g}, s)$ is scored against a probabilistic TL model M_{TL};

[2] Each SL segment s is analyzed using the morphological analyzer of the MT system; for each SL word the set of possible PoS tags is obtained.

– the probability $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ of each translation $\tau(\boldsymbol{g}, s)$ in the language model M_{TL} is used to estimate the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ of each disambiguation $\boldsymbol{g}$ given the SL segment s in the tagging model M_{tag} we are trying to learn; and, finally,
– the estimated probabilities $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ are used to determine the parameters of the tagging model M_{tag} by using them as partial counts, that is, as if disambiguation $\boldsymbol{g}$ had been seen $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ times in the training corpus for the SL segment s.

The following example illustrates how the method works. Suppose that we are training the English PoS tagger to be used within an RBMT system translating from English to Spanish, and that we have the following segment in English, $s =$*He books the room*. The first step is to use the morphological analyzer of the MT system 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 with two possible PoS tags each (*books* and *room*) we have, for the given segment, four disambiguation choices or PoS combinations:

– $\boldsymbol{g}_1 =$ (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 English (SL) segment into Spanish (TL) according to each disambiguation $\boldsymbol{g}$:

– $\tau(\boldsymbol{g}_1, s) =$*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*.

Then each translation $\tau(\boldsymbol{g}, s)$ is scored against a Spanish language model M_{TL}. It is expected that a reasonable Spanish language model M_{TL} will give a higher likelihood $P_{\mathrm{TL}}(\tau(\boldsymbol{g}_1, s))$ to $\tau(\boldsymbol{g}_1, s)$ than to the remaining translations ($\tau(\boldsymbol{g}_2, s)$, $\tau(\boldsymbol{g}_3, s)$ and $\tau(\boldsymbol{g}_4, s)$), as they make little sense in Spanish. In the method presented here, the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ of each tag sequence $\boldsymbol{g}$ given the SL segment s in the tagging model M_{tag} is taken to be proportional to the likelihood $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ of their respective translations into TL and then used to estimate the HMM parameters.

As the previous example illustrates, the method uses an untagged SL corpus as input, and a TL model M_{TL}. The input SL corpus must be *segmented* before training in order to consider all disambiguations for each segment independently of the others. In this work, a *segment* is a sequence of words that is processed independently of the adjacent segments by the MT modules following the PoS tagger. Concerning the TL model, in this paper we consider a classical trigram language model based on surface forms (words as they appear in raw corpora).

We have previously described this unsupervised method (Sánchez-Martínez et al. 2004a,b). Here we introduce some changes in the method used in those papers, give a stronger mathematical motivation, report better results, discuss confidence intervals, and evaluate the method on three different SLs (Spanish, Occitan and French), all of them being translated into Catalan with the open-source shallow-transfer MT system

Apertium (see Appendix A). We also report results when the structural (syntactic) transfer component is removed from the MT system and substituted by a "null" (context-free word-for-word) structural transfer module. Finally, we test a simple technique (Sánchez-Martínez et al. 2006) on the three mentioned languages, which may be used to reduce the number of translations per segment during training without degrading the accuracy achieved by the resulting PoS tagger. Note that the translation of all possible disambiguations of each segment is the most time-consuming task of the proposed method, and avoiding translating a large portion of them makes the training method much faster.

The rest of the paper is organized as follows. The next section gives a formal description of the method discussed in this paper; then, Sect. 3 introduces a simple pruning method that can be used to speed up the training method by avoiding a significant number of translations. In Sect. 4, the requirements of the segmentation algorithm are discussed. Section 5 presents the various sets of experiments conducted. Finally, in Sects. 6 and 7 we discuss the results and outline future work to be done.

2 A machine translation-oriented HMM training method

This section presents the mathematical details of the new unsupervised method to train SL HMM-based PoS taggers to be used in MT introduced in the previous section; as the goal is to train PoS taggers for their use in MT, this new training method will be referred as an *MT-oriented* method.[3] Despite the fact that information in the rest of the modules of the MT system is used, this training method may be said to be unsupervised because no hand-tagged corpora are needed.

Every HMM training algorithm calculates, if possible, or estimates the frequency counts $n(\cdot)$ from which the HMM parameters are estimated. What follows is the mathematical justification that allows the estimation of these frequency counts from statistics collected from TL corpora.

Learning an SL PoS tagging model M_{tag} using information from both TL and SL by means of an MT system can be seen as trying to approximate Eq. (1):

$$P_{\text{TL}}(t) \simeq P_{\text{trans,tag,SL}}(t) \tag{1}$$

that is, approximating the probability $P_{\text{TL}}(t)$ of every TL segment t in a TL model M_{TL} as the probability of t in a composite model consisting of a translation model M_{trans}, the PoS tagger model M_{tag} whose parameters we are trying to learn, and an SL model M_{SL}.

As the goal is to learn the parameters of the SL PoS tagger model M_{tag}, special attention must be paid to all possible disambiguations (PoS tag sequences). Taking this into account, and the way in which a TL segment t would be produced from an

[3] The method described in this section is implemented inside the open-source package `apertium-tagger-training-tools` released under the GNU GPL license; it can be freely downloaded from http://sf.net/projects/apertium.

SL segment s through an MT model M_{trans} and an SL PoS tagger M_{tag}, the right-hand side of Eq. (1) can be rewritten as in (2):

$$P_{\text{trans,tag,SL}}(t) = \sum_{s} \sum_{\boldsymbol{g}} P_{\text{trans}}(t|\boldsymbol{g}, s)\ P_{\text{tag}}(\boldsymbol{g}|s)\ P_{\text{SL}}(s) \tag{2}$$

where $\boldsymbol{g} = (\gamma_1 \ldots \gamma_N)$ is a sequence of PoS tags in the SL; $P_{\text{trans}}(t|\boldsymbol{g}, s)$ is the probability in the translation model M_{trans} of a TL segment t given a tag sequence $\boldsymbol{g}$ and an SL segment s; $P_{\text{tag}}(\boldsymbol{g}|s)$ is the probability in the PoS tagger model M_{tag} of the tag sequence $\boldsymbol{g}$ given the source segment s; and $P_{\text{SL}}(s)$ is the probability in the SL model M_{SL} of that SL segment. The unrestricted sums over all possible s and over all possible tag sequences $\boldsymbol{g}$ are necessary until we know more about the models.

Once we have the general equation describing how the tagging model M_{tag} is related to the TL within an MT system, we can make some choices regarding the models being used. We have chosen our translation model to be a rule-based system which assigns a single TL segment $\tau(\boldsymbol{g}, s)$ to each source segment s and PoS tag sequence $\boldsymbol{g}$; therefore, we can write (3):

$$P_{\text{trans}}(t|\boldsymbol{g}, s) = \delta_{t,\tau(\boldsymbol{g},s)} \tag{3}$$

where $\delta_{a,b}$ is the Kronecker delta ($\delta_{a,b} = 1$ if $a = b$ and zero otherwise).[4] Thus, our basic equation can be then rewritten to integrate the translation model as in (4):

$$P_{\text{trans,tag,SL}}(t) = \sum_{s} \sum_{\boldsymbol{g}} \delta_{t,\tau(\boldsymbol{g},s)} P_{\text{tag}}(\boldsymbol{g}|s)\ P_{\text{SL}}(s) \tag{4}$$

We have also chosen the PoS tagging model M_{tag} to be an HMM $\lambda = (\Gamma, \Sigma, A, B, \pi)$, in where Γ refers to the set of hidden states (PoS tags), Σ refers to the set of observable outputs (word classes), and A, B and π to the transition probabilities, emission probabilities and initial probabilities, respectively. Appendix B gives a brief explanation of how HMMs are used to perform PoS tagging and the assumptions made to avoid learning the probability π of each PoS tag being the initial one.

As a consequence of the tagging model M_{tag}, the set $T(s)$ of PoS tag sequences $\boldsymbol{g}$ that can be assigned to a source segment s is finite, and equal to all possible PoS tag combinations of words in s. Because of this we will call each $\boldsymbol{g}$ a *path* since it describes an unique state path in the HMM.

At this point, Eq. 4 can be rewritten as in (5):

$$P_{\text{trans,tag,SL}}(t) = \sum_{s:\ \tau(\boldsymbol{g},s)=t,\ \boldsymbol{g}\in T(s)} P_{\text{tag}}(\boldsymbol{g}|s)\ P_{\text{SL}}(s) \tag{5}$$

where the translation model has been integrated as a restriction over summations.

[4] A different model could be used; for instance, one where a segment s tagged as $\boldsymbol{g}$ could have more than one translation ("polysemy"), that is, one where $\tau(\boldsymbol{g}, s)$ is a set. This model would have additional parameters that would have to be known or trained.

Now that we have integrated the particular PoS tagging model M_{tag} to be learned and the translation model M_{trans} to be used, we need to make some approximations and assumptions in order to make the method work in a practical framework.

Approximations and assumptions. As the translation model M_{trans} has no analytical form and the number of possible SL segments s are in principle infinite, it is unfeasible to solve Eq. (5) for all possible $P_{\text{tag}}(\boldsymbol{g}|s)$, even if an SL model M_{SL} is available. Therefore, our method will take *samples* from representative SL texts; that is, a representative SL corpus will be used as a *source* of segments to process (approx. #1). An additional approximation here is this; when computing the contribution of each segment s to the HMM parameters A and B, the possible contributions of other SL segments s' to the same translation t may be safely *ignored* (approx. #2); that is, it is assumed that it is unlikely that a segment s' has for some disambiguation $\boldsymbol{g}'$ the same translation that s has.

Applying it to a single sampled segment s, Eq. (5) may be written as in (6):

$$P_{\text{trans,tag}}(t|s) = \sum_{\tau(\boldsymbol{g},s)=t,\ \boldsymbol{g}\in T(s)} P_{\text{tag}}(\boldsymbol{g}|s) \tag{6}$$

where the probability of a TL segment t given the SL segment s is computed as the sum over all disambiguations $\boldsymbol{g} \in T(s)$ of the probability of each $\boldsymbol{g}$, given the source segment s and the HMM M_{tag} we are trying to learn.

The main assumption in this work is that the probability $P_{\text{trans,tag}}(t|s)$ can be approximated (approx. #3) through a TL model, as in (7):

$$P_{\text{trans,tag}}(t|s) \simeq \begin{cases} \frac{1}{k_s}\ P_{\text{TL}}(t) & \text{if } \exists \boldsymbol{g}:\ \tau(\boldsymbol{g},s)=t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

with

$$k_s = \sum_{t':\ (\exists \boldsymbol{g}:\ \tau(\boldsymbol{g},s)=t')} P_{\text{TL}}(t') \tag{8}$$

where k_s is the sum of the probabilities of all possible translations into TL of the SL segment s according to all the disambiguations given by $T(s)$; that is, the probabilities of TL sentences t that cannot be produced as a result of the translation of SL segment s by means of the MT system being used are not taken into account.

At this point we have two different ways of computing the probability $P_{\text{trans,tag}}(t|s)$. Making both right-hand sides of Eqs. (6) and (7) equal when $\tau(\boldsymbol{g}, s) = t$ yields (9):

$$\sum_{\substack{\boldsymbol{g}'\in T(s),\\ \tau(\boldsymbol{g}',s)=\tau(\boldsymbol{g},s)}} P_{\text{tag}}(\boldsymbol{g}'|s) \simeq \frac{1}{k_s}\ P_{\text{TL}}(\tau(\boldsymbol{g}, s)) \tag{9}$$

where t has been replaced by $\tau(\boldsymbol{g}, s)$ because of the restriction over t introduced by the translation model M_{trans}. From now on we will use $\tau(\boldsymbol{g}, s)$ instead of t to mean that this restriction holds.

As can be seen in Eq. (9) more than one $\boldsymbol{g}$ may contribute to the same translation $\tau(\boldsymbol{g}, s)$. The following example illustrates this phenomenon. Suppose the French segment *La ville* and that an MT system translating from French to Spanish is available. The morphological analysis according to the lexicon of this segment is: *La* (article or pronoun) *ville* (noun). This segment has only two disambiguation paths, $\boldsymbol{g}_1$ = (article, noun) and $\boldsymbol{g}_2$ = (pronoun, noun), but the translation into Spanish is the same (*La ciudad*) for both paths since word *La* is involved in a *free ride*, a phenomenon by which choosing the incorrect interpretation for an ambiguous word does not result in a translation error. The more related two languages are, the more frequent this free-ride phenomenon is.[5]

Let $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s)$ be a factor that measures the (fractional) contribution of disambiguation $\boldsymbol{g}$ to the translation into TL $\tau(\boldsymbol{g}, s)$ of segment s, that is, $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s)$ dictates how the probability $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ must be shared out, after normalization, between all the disambiguation paths of segment s producing $\tau(\boldsymbol{g}, s)$. At this point we can then rewrite Eq. (9) as in (10):

$$P_{\mathrm{tag}}(\boldsymbol{g}|s) \simeq \frac{1}{k_s} P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))\, \xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s) \tag{10}$$

The fact that more than one path in segment s, say $\boldsymbol{g}$ and $\boldsymbol{g}'$, produce the same translation $\tau(\boldsymbol{g}, s)$ does not necessarily imply that $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}_i, s), s) = \xi(\boldsymbol{g}', \tau(\boldsymbol{g}_i, s), s)$. However, in the absence of further information, the contributions of each path will be approximated as being equal (approx. #4), as in (11):

$$\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s) \approx \frac{1}{\mid \{\boldsymbol{g}' \in T(s) : \tau(\boldsymbol{g}', s) = \tau(\boldsymbol{g}, s)\} \mid} \tag{11}$$

Although this approximation may affect PoS tagging performance, it is expected to affect translation quality very indirectly; remember that the method presented is aimed at training PoS taggers to be used in MT; therefore, what really matters is translation quality, not tagging accuracy.

Integrating Eq. (11) into Eq. (10) we have (12):

$$P_{\mathrm{tag}}(\boldsymbol{g}|s) \simeq \frac{1}{k_s} \frac{P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))}{\mid \{\boldsymbol{g}' \in T(s) : \tau(\boldsymbol{g}', s) = \tau(\boldsymbol{g}, s)\} \mid} \tag{12}$$

which expresses a proper probability model as can be easily shown by summing over all possible disambiguation paths $\boldsymbol{g}$ of SL segment s.

Equation (12) shows how a given disambiguation $\boldsymbol{g}$ of words in an SL segment s is related to the TL. Thus the values of $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ approximated in this way can be used

[5] Table 3 gives an idea on how frequent this phenomenon is in the language pairs considered in our experiments.

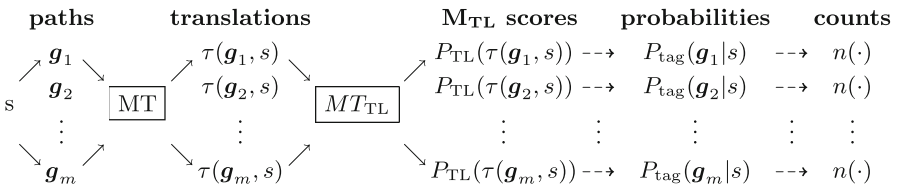

Fig. 2 Scheme of the process followed by the method to estimate the frequency counts $n(\cdot)$ needed to obtain the HMM parameters. These counts are based on the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ of each disambiguation $\boldsymbol{g}$ in the tagging model M_{tag} given SL segment s

as fractional counts to estimate the HMM parameters A and B so that Eq. 12 holds as close as possible.

The objective is to estimate the frequency counts $n(\cdot)$ needed to estimate the HMM parameters by using the approximate probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ as an information source.

Figure 2 summarizes the process followed to estimate the frequency counts needed. First of all the input SL text is segmented, and all possible disambiguation paths $\boldsymbol{g}$ for each segment are considered. Therefore, for a given segment s the translations $\tau(\boldsymbol{g}, s)$ of segment s according to each possible disambiguation $\boldsymbol{g} \in T(s)$ are performed. Once all different translations of segment s have been obtained, each translation $\tau(\boldsymbol{g}, s)$ is scored using the target-language model M_{TL}. Then these scores are used to estimate the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ in the tagging model M_{tag} of each path $\boldsymbol{g}$ being the correct disambiguation of segment s using Eq. (12). Finally, these probabilities are used to estimate the frequency counts we have already mentioned, as we now describe.

The frequency counts $n(\cdot)$ from which the HMM parameters are estimated can be obtained from the estimated probabilities $P_{\mathrm{tag}}(\boldsymbol{g}|s)$; in this case, frequency counts are approximations $\tilde{n}(\cdot)$, rather than exact values $n(\cdot)$. In order to approximate these counts, each $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ is treated as a fractional count, i.e. as if the disambiguation $\boldsymbol{g}$ of segment s had been seen $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ times. An estimate of tag occurrences based on $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ is given in (13):

$$\tilde{n}(\gamma_i) \cong \sum_{n=1}^{N_S} \sum_{\boldsymbol{g} \in T(s_n)} C_{s_n,\boldsymbol{g}}(\gamma_i)\, P_{\mathrm{tag}}(\boldsymbol{g}|s_n) \tag{13}$$

where N_S is the number of segments in the SL training corpus, and $C_{s_n,\boldsymbol{g}}(\gamma_i)$ is the number of times tag γ_i appears in path $\boldsymbol{g}$ of segment s_n.

Analogously, an estimate of the tag pair occurrence frequency based on $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ is given in (14):

$$\begin{aligned}\tilde{n}(\gamma_i\gamma_j) \cong & \sum_{n=1}^{N_S} \sum_{\boldsymbol{g} \in T(s_n)} C_{s_n,\boldsymbol{g}}(\gamma_i, \gamma_j)\, P_{\mathrm{tag}}(\boldsymbol{g}|s_n) \\ & + \sum_{n=1}^{N_S} \left(\sum_{\substack{\boldsymbol{g}' \in T(s_{n-1}),\\ \gamma_i = \mathrm{last}(\boldsymbol{g}')}} P_{\mathrm{tag}}(\boldsymbol{g}'|s_{n-1}) \sum_{\substack{\boldsymbol{g} \in T(s_n),\\ \gamma_j = \mathrm{first}(\boldsymbol{g})}} P_{\mathrm{tag}}(\boldsymbol{g}|s_n) \right)\end{aligned} \tag{14}$$

$s \equiv$	*He*	*books*	*the*	*room*	
	{ PRN }	{ VB NN }	{ ART }	{ VB NN }	
					$P_{\text{tag}}(\boldsymbol{g}\|s)$
$\boldsymbol{g}_1 \equiv$	PRN	VB	ART	NN	
$\tau(\boldsymbol{g}_1, s) \equiv$	*Él*	*reserva*	*la*	*habitación*	0.75
$\boldsymbol{g}_2 \equiv$	PRN	VB	ART	VB	
$\tau(\boldsymbol{g}_2, s) \equiv$	*Él*	*reserva*	*la*	*aloja*	0.15
$\boldsymbol{g}_3 \equiv$	PRN	NN	ART	NN	
$\tau(\boldsymbol{g}_3, s) \equiv$	*Él*	*libros*	*la*	*habitación*	0.06
$\boldsymbol{g}_4 \equiv$	PRN	NN	ART	VB	
$\tau(\boldsymbol{g}_4, s) \equiv$	*Él*	*libros*	*la*	*aloja*	0.04

Fig. 3 Example of an ambiguous SL (English) segment s, paths and translations $\tau(\boldsymbol{g}, s)$ into TL (Spanish) resulting from each possible disambiguation $\boldsymbol{g}$, and estimated probability $P_{\text{tag}}(\boldsymbol{g}|s)$ of each path being the correct disambiguation

where $C_{s_n,\boldsymbol{g}}(\gamma_i, \gamma_j)$ is the number of times tag γ_i is followed by tag γ_j in path $\boldsymbol{g}$ of segment s_n, and first($\boldsymbol{g}$) and last($\boldsymbol{g}$) are two functions returning the first and last tag, respectively, of the disambiguation path $\boldsymbol{g}$. Note that the second term of the addition considers the boundary between two adjacent segments.

The number of times a word class σ_k is emitted by a given tag γ_j is approximated as in (15):

$$\tilde{n}(\sigma_k, \gamma_j) \cong \sum_{n=1}^{N_S} \sum_{\boldsymbol{g} \in T(s_n)} C_{s_n,\boldsymbol{g}}(\sigma_k, \gamma_j)\, P_{\text{tag}}(\boldsymbol{g}|s_n) \tag{15}$$

where $C_{s_n,\boldsymbol{g}}(\sigma_k, \gamma_j)$ is the number of times word class σ_k is emitted by tag γ_j in path $\boldsymbol{g}$ of segment s_n.

Finally, note that the number of times that the ambiguity class σ_k appears in the training corpus $n(\sigma_k)$ does not need to be approximated, as it can be easily computed from the untagged training corpus.

Figure 3 outlines the application of the method to an isolated segment when a TL language model M_{TL} based on trigrams of words is used.

3 Pruning of disambiguation paths

In this section we focus on the main disadvantage of the training method presented, namely the large number of translations that have to be performed for each segment, and how to alleviate this problem. The objective of the method introduced in this section is to reduce as much as possible the number of translations to perform per

segment without degrading the translation performance achieved by the MT system embedding the resulting PoS tagger.

3.1 Pruning method

The disambiguation pruning method is based on a priori knowledge, i.e. on an initial model $\hat{M}_{\text{tag}}^{[0]}$ of SL tags. The assumption here is that any reasonable model of SL tags may be helpful in choosing a subset of possible disambiguation paths, such that the correct one is contained in that subset. Therefore, there is no need to translate all possible disambiguation paths of each segment into the TL, but only the most *promising* ones.

The initial model $\hat{M}_{\text{tag}}^{[0]}$ of SL tags can be either an HMM or any other model whose parameters are obtained by means of a statistically sound method. Nevertheless, using an HMM as an initial model allows the method to dynamically evolve, obtaining a new model $\hat{M}_{\text{tag}}$ that is the result of integrating new evidence collected during training (see Sect. 3.3 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 $\hat{P}_{\text{tag}}(\boldsymbol{g}|s)$ of each possible disambiguation path $\boldsymbol{g}$ of segment s in the tagging model $\hat{M}_{\text{tag}}$ is calculated; then, the subset of disambiguation paths to be taken into account is determined according to the calculated a priori likelihoods.

Let $U(s)$ be an ordered set of all possible disambiguation paths of the SL segment s; disambiguation paths $\boldsymbol{g} \in U(s)$ are ordered in decreasing order of their a priori likelihood, that is, $U(s) = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{|T(s)|}\}$ with $\boldsymbol{g}_i \in T(s) : \ 1 \leq i \leq |T(s)|$, and $\hat{P}_{\text{tag}}(\boldsymbol{g}_i|s) \geq \hat{P}_{\text{tag}}(\boldsymbol{g}_{i+1}|s)$.

To decide which disambiguation paths to take into account, the pruning algorithm is controlled by a mass probability threshold $\rho \in [0, 1]$; the subset of disambiguation paths to take into account, $U'(s) = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k\}$ with $k \leq |T(s)|$, must satisfy the expression in (16):

$$\rho \leq \sum_{i=1}^{k} \hat{P}_{\text{tag}}(\boldsymbol{g}_i|s) \tag{16}$$

for the minimum possible value of k. Therefore, after pruning, the training method described in Sect. 2 takes into account the minimum subset of disambiguation paths $\boldsymbol{g} \in T(s)$ needed to reach the mass probability threshold ρ. Note that the disambiguation paths $\boldsymbol{g}$ that have not been taken into account will be assumed to have a null $P_{\text{tag}}(\boldsymbol{g}|s)$ when estimating the frequency counts $\tilde{n}(\cdot)$ via Eqs. (13)–(15).

3.2 Estimation of the a priori likelihood

The estimation of the a priori likelihood $\hat{P}_{\text{tag}}(\boldsymbol{g}|s)$ of each disambiguation path $\boldsymbol{g}$ is done by taking into account the context in which segment s appears. Context needs to be taken into account as a consequence of the segmentation strategy because, on the

one hand, segments may start at words that would never appear at the beginning of a well-formed sentence, which makes using the vector π with the probability of each PoS tag being the initial one completely inadequate, and, on the other hand, because some segments may be too short to estimate an accurate a priori likelihood.

Context is taken into account by calculating the forward and backward probabilities, as in the Baum-Welch EM algorithm (see Eqs. 1 and 2 in Cutting et al. 1992). After that, the a priori likelihood of disambiguation path $\boldsymbol{g} = (\gamma_1 \ldots \gamma_N)$ given segment $s = (\sigma_1 \ldots \sigma_N)$ is calculated using Eq. (17):

$$\hat{P}_{\text{tag}}(\boldsymbol{g}|s) = \sum_{\gamma_j \in \Gamma} \alpha_-(\gamma_j) a_{\gamma_j \gamma_1} b_{\gamma_1}(\sigma_1) \prod_{i=2}^{N} a_{\gamma_{i-1}\gamma_i} b_{\gamma_i}(\sigma_i) \times \sum_{\gamma_j \in \Gamma} a_{\gamma_N \gamma_j} b_{\gamma_j}(\sigma_j) \beta_+(\gamma_j) \tag{17}$$

where $\alpha_-(\gamma_j)$ and $\beta_+(\gamma_j)$ refers to the forward probability of PoS tag γ_j for the word preceding the first one in the segment being considered, and to the backward probability of PoS tag γ_j for the first word after the last one of segment s, respectively.

3.3 HMM updating

This section explains how the model $\hat{M}_{\text{tag}}$ used for pruning can be updated during training so that it integrates new evidence collected from the TL. The idea is to periodically estimate an HMM using the counts collected from the TL (as explained in Sect. 2), and to mix the resulting HMM with the initial one; the mixed HMM becomes the new model $\hat{M}_{\text{tag}}$ used for pruning.

The initial model and the model obtained during training are mixed so that the estimate of a priori likelihoods is the best possible at each moment; mixing affects both transition and emission probabilities.

Let $\boldsymbol{\theta} = (a_{\gamma_1\gamma_1}, \ldots, a_{\gamma_{|\Gamma|}\gamma_{|\Gamma|}}, b_{\gamma_1}(\sigma_1), \ldots, b_{\gamma_{|\Gamma|}}(\sigma_{|\Sigma|}))$ be a vector containing all of the parameters of a given HMM. The mixing of the initial HMM and the new one can be achieved via the linear combination in (18):

$$\boldsymbol{\theta}(x) = \varphi(x)\, \boldsymbol{\theta}^{\text{TL}}(x) + (1 - \varphi(x))\, \boldsymbol{\theta}^{[0]} \tag{18}$$

where $\boldsymbol{\theta}(x)$ refers to the HMM parameters after mixing the two models when a fraction x of the training corpus has been processed; $\boldsymbol{\theta}^{\text{TL}}(x)$ refers to the HMM parameters estimated by means of the MT-oriented method described in Sect. 2 after processing a fraction x of the SL training corpus; and $\boldsymbol{\theta}^{[0]}$ refers to the parameters of the initial HMM. The function $\varphi(x)$ assigns a weight to the model estimated using the counts collected from the TL ($\boldsymbol{\theta}^{\text{TL}}$). This monotonically increasing weight function is made to depend on the fraction x of the SL corpus processed so far so that $\varphi(0) = 0$ and $\varphi(1) = 1$.

4 Segmenting the SL text

In the previous sections we have discussed segments as the SL units to be processed by the method. In the introduction we defined a segment as a sequence of words that is processed independently of the adjacent segments by the remaining modules of the MT system after the PoS tagger. SL text segmentation must indeed be carefully designed so that two words which are jointly treated at some stage in the MT process after the PoS tagger are not assigned to different segments. This would result in incorrect sequences in the TL (for example, if two words involved in a word-reordering or agreement rule are assigned to different segments) and, as a consequence of that, in wrong likelihood estimations. However, it must be noted that, when related languages are involved, even if segment independence is not guaranteed, a large fraction of segment translations may be still correct because of the small grammatical divergences between the languages involved (see Sect. 5.6).

Using whole sentences as segments seems to be a reasonable choice, because most current MT systems translate at sentence level, with each sentence translated independently of any other SL sentence. However, since the number of disambiguations grows exponentially with sentence length, we need to segment sentences in order to make the problem computationally feasible. In general, first-order HMMs can be trained by breaking the corpus into segments whose first and last words are unambiguous in the same way that the Viterbi algorithm (Rabiner 1989; Manning and Schütze 1999, p. 332) is used for disambiguation. Adequate strategies for ensuring segment independence depend on the particular translation system. In Sects. 5.5 and 5.6, the strategy used in each experiment will be described.

5 Experiments

The method we present is aimed at producing PoS taggers to be used in MT; to test this new approach we used the Apertium open-source shallow-transfer MT platform (see Appendix A) and data for the translation from three different languages—Spanish, French and Occitan—into Catalan. Note that when training the PoS taggers the whole MT engine, except the PoS tagger, is used to produce all the translations $\tau(\boldsymbol{g}, s)$ that are evaluated through the TL model M_{TL}; because of this we say that the MT-oriented method also uses information in the remaining modules of the MT engine. As a TL model we used a classical trigram language model trained from a raw-text Catalan corpus consisting of around 2 million words. Trigram probabilities were smoothed by means of the *deleted interpolation* (Jelinek 1997, ch. 4) method in conjunction with the *successive linear abstraction* approximation (Brants and Samuelsson 1995) to compute smoothing coefficients and the Good-Turing method (Gale and Sampson 1995) to smooth unigram probabilities.[6]

[6] A possible criticism may be that we have used our own language model implementation instead of a well-known toolkit such as SRILM (Stolcke 2002); this is because the SRILM toolkit is only available for non-profit purposes and we wanted our method to be freely accessible under a standard open-source license to everyone (in particular, academia and industry). In any case, the choice of a particular language model implementation is orthogonal to our method; experiments conducted when training a Spanish PoS tagger

Table 1 Main figures for the tagsets used by the corresponding PoS tagger for each language

Language	Fine tags (analyzer)				Coarse tags (tagset)			
	Single-word	Multi-word	$\|\Gamma\|$	$\|\Sigma\|$	Single-word	Multi-word	$\|\Gamma\|$	$\|\Sigma\|$
Spanish	375	1,739	2,114	2,640	85	14	99	291
French	318	102	420	741	72	4	76	264
Occitan	346	1,957	2,303	2,930	87	18	105	345

Each tagset consists of a set of coarse tags grouping together the finer tags delivered by the morphological analyzer. There are single-word tags and multi-word tags. Multi-word tags are used for SL contractions and verbs with attached clitics. Grouping fine tags into coarse tags reduces the total number of states $|\Gamma|$ and the number of word classes $|\Sigma|$ that need to be taken into account

The next section describes the tagset used for each language; then, the different corpora used for training and the corpora used for evaluation are described in Sect. 5.2. Section 5.3 describes the measures used to evaluate the MT-oriented approach; then, Sect. 5.4 provides an explanation of the different MT setups used as references. After that, we provide the results obtained by our (MT-oriented) method when training PoS taggers for Spanish, French and Occitan for translation into Catalan. Firstly, in Sect. 5.5 we report the results when a complete structural transfer module is used to produce all the translations $\tau(\boldsymbol{g}, s)$; then, in Sect. 5.6, we present the results achieved when the structural transfer module (see Appendix A) is simplified to a minimum (context-free word-for-word) 'null' structural transfer model. Finally, in Sect. 5.7, we report the results achieved when applying the path pruning technique described in Sect. 3.

5.1 Tagset

The tagset used by the corresponding PoS tagger for each language consists of a set of coarse tags which group together the finer tags generated by the morphological analyzer. Table 1 summarizes the main features of the three tagsets used. The number of word classes $|\Sigma|$ is also given. When defining the tagset (see Sect. 7 for more details) a few very frequent ambiguous words are assigned special hidden states (Pla and Molina 2004), and consequently special word classes. In our Spanish tagset only the words *para* (preposition or verb), *que* (conjunction or relative), *como* (preposition, relative or verb), *algo* (pronoun or adverb), and *más/menos* (adverb or adjective) are assigned special hidden states in Γ; for Occitan, words *que* (conjunction or relative), *molt* (adjective or adverb), *a* (preposition or verb), and *auer* (verb) are also assigned special hidden states; for French no special hidden states are used.

5.2 Corpora

The corpora used for training and testing come from different newspapers and institutional web pages.

Footnote 6 continued
through our MT-oriented method using the well-known SRILM toolkit (trained with the following options: `-order 3 -interpolate -kndiscount -unk`) provide results which are indistinguishable in practice from those reported in this paper.

Table 2 Figures regarding training corpora: number of words, vocabulary size, percentage of ambiguous words (PoS-amb., without considering unknown words), percentage of words with more than one translation into Catalan due to PoS ambiguities (non-free PoS-amb.), and percentage of unknown words

Language	#*Words*	Vocab. size	PoS-amb. (%)	Non-free PoS-amb. (%)	Unk. words (%)
Spanish	500,072	43,789	23.32	7.20	4.16
French	500,083	44,374	28.11	17.06	9.48
Occitan	300,034	28,929	27.50	19.98	4.70

5.2.1 Training

With the aim of testing whether the amount of training corpora needed for convergence is consistent across different experiments, and to see whether the method behaves in the same way in terms of performance, for all experiments we used 5 disjoint corpora for both Spanish and French, and only one corpus for Occitan.[7] The Spanish training corpora come from the Spanish newspaper *El País*;[8] the French training corpora come from the newspaper *Le Monde*;[9] finally, the Occitan training corpus comes from a weekend supplement in Occitan of the Catalan newspaper *Avui*[10] published during 2002.

Table 2 reports, for the three languages considered in our experiments, the number of words, the size of the vocabulary (number of distinct words), the percentage of words which are ambiguous because of having more than one possible PoS (*PoS-ambiguous* words, PoS-amb.), the percentage of words with more than one translation into Catalan because of PoS ambiguities (non-free PoS-ambiguous words, non-free PoS-amb.), and the percentage of words in each training corpus that are unknown to the MT system used in the experiments. Note that in the case of Spanish and French, Table 2 only reports data for one of the training corpora used, as the rest of the corpora show similar values.

The reported data gives an idea of the ambiguity found in the training corpora and the incidence of the *free ride* phenomenon (see Sect. 2). Note, however, that the percentage of non-free PoS-ambiguous words must be interpreted as a lower bound to the percentage of MT errors that may be produced if all PoS ambiguities are incorrectly solved. It is a lower bound because, under some circumstances, the remaining PoS-ambiguous words (free PoS-ambiguous words) may still cause neighbouring words to be incorrectly translated if incorrectly tagged because of differences regarding how transfer rules are activated; therefore, PoS-ambiguous words that are free are not always involved in a *free-ride* phenomenon.

[7] Note that Occitan has a reduced community of native speakers (about one million people), and that it is legally recognized only in the Val d'Aran (a small valley of the Pyrenees of Catalonia), where it is official (with some limitations) together with Catalan and Spanish. In addition, Occitan dialects have strong differences, and its standardization as a unified language still faces a number of open issues.

[8] http://www.elpais.com.

[9] http://www.lemonde.fr.

[10] http://www.avui.es.

Table 3 Figures regarding evaluation corpora: number of SL words, vocabulary size, number of sentences, percentage of ambiguous words (PoS-amb., without considering unknown words), percentage of words with more than one translation into Catalan due to PoS ambiguities (non-free PoS-amb.), and percentage of unknown words

Language	# Words	Vocab. size	# Sent.	PoS-amb. (%)	Non-free PoS-amb. (%)	Unk. words (%)
Spanish	10,066	3,276	457	23.03	6.36	4.90
French	10,154	3,343	387	29.32	17.06	10.35
Occitan	10,079	3,314	538	30.63	21.69	4.97

In the case of Spanish a large portion of the PoS-ambiguous words found in the training corpus are free PoS-ambiguous; this is explained by the fact that the second most frequent word in Spanish (*la*), which accounts for 3.60% of the training corpus, has two possible PoS tags which are both translated in the same way into Catalan, except under certain circumstances due to the effect of a structural transfer rule ensuring noun-phrase agreement.

5.2.2 Testing

The method we present to train HMM-based PoS taggers for use in MT is evaluated in the following sections by observing the translation performance of the MT system embedding the resulting PoS tagger when translating a test corpus independent of the one used for training. The Spanish test corpus comes from the newspaper *20 Minutos*;[11] the French test corpus comes from the *Portal Turístic d'Andorra*,[12] the official web page about tourism in Andorra; the Occitan test corpus comes from *Aran ath Dia*, a magazine that is published monthly with articles and news regarding daily life in the Val d'Aran. For each language only one reference translation was used, which was built by post-editing the MT output into Catalan performed with the same MT system and linguistic data.

Table 3 shows, for the different SL corpora used for evaluation, the same data reported for the training corpora plus the number of sentences in each test corpus. Note that the percentages reported are in line with those reported for the training corpora.

In the case of Spanish we also evaluate the PoS tagging performance in view of the possible applications of the resulting PoS tagger in other NLP applications; unfortunately, this evaluation could not be done in the case of the other two languages because no hand-tagged corpora were available.

The Spanish PoS tagging error rates are evaluated using an independent Spanish hand-tagged corpus, consisting of 253 sentences and 8,059 words from the Spanish newspaper *El País*. In this corpus the percentage of ambiguous words according to the lexicon, including unknown words, is 27.6% (3.9% unknown, 23.7% known). Note that when evaluating via this tagged corpus, 0.8% of the words are always incorrectly tagged since the correct PoS tag (in the evaluation corpus) is never provided by the morphological analyzer due to incomplete morphological entries in the lexicon.

[11] http://www.20minutos.es.

[12] http://www.andorra.ad.

5.3 Machine translation evaluation

Translation performance was evaluated using two different measures: word error rate (WER) and BLEU (Papineni et al. 2002). The translation edit rate (TER) (Snover et al. 2006), which works very similarly to WER but allows for phrasal (block) shifts, was also used to evaluate translation performance. However, we do not report results for TER as the scores obtained were indistinguishable in practice from the corresponding WER figures. This is because the number of shifts to perform when correcting the MT output was negligible, given that the languages involved in the translation are closely related.

WERs are computed as the word-level edit distance (Levenshtein 1965) between the translation being evaluated and the reference translation. As the reference translation used for each language is a post-edited machine translation performed using the same data (see Sect. 5.2.2), the WER gives an idea of how much each method helps human translators in their daily work, since it provides the percentage of words that need to be inserted, replaced or deleted to transform the MT output into an adequate translation into TL. Concerning the BLEU metric, it must be noted that as only one reference translation is used, and that the reference is a human-corrected version of the same MT output, BLEU scores are higher than might initially be expected; in any case, what really counts is how much the reported scores vary. It is worth noting that as the rest of the MT modules are the same for all PoS taggers tested, the differences in translation performance are due solely to changes in the output produced by the PoS tagger module.

Confidence intervals. To allow for an easier interpretation and to permit a better comparison between the quality measures, in the following sections we report each performance value together with its confidence interval.

Confidence intervals of quality measures are calculated using *bootstrap resampling* (Koehn 2004). In general, bootstrap resampling consists of estimating the precision of sample statistics (in our case, translation or PoS tagging quality measures) by randomly resampling with replacements (that is, allowing repetitions) from the full set of samples (Efron and Tibshirani 1993) (in MT, sentences and their respective reference translations). This method has the property that no assumptions are made about the underlying distribution of the variable: in our case, the corresponding quality measure.

The calculation of the confidence intervals consists of the following steps:

1. the quality measure is calculated a large number of times using randomly chosen sentences from the test corpus, and their counterpart sentences in the reference corpus;
2. all the calculated measures are sorted in ascending order; and
3. the top $q\%$ and the bottom $q\%$ elements are removed from that list.

After that, the remaining values are in the interval $[a, b]$. This interval approximates with probability $1 - 2q/100$ the range of values within which the quality measure lies for evaluation corpora with a number of sentences equal to that used to carry out the evaluation (see Table 3). In the following sections we report the centre of the interval together with its width.

5.4 Reference results

In the following sections the performance of our method to train HMM-based PoS taggers for MT is evaluated on three different languages—Spanish, French and Occitan—being translated into Catalan. Here we describe the MT setups used as references:

Baum-Welch: The HMM-based PoS tagger is trained by following the classical unsupervised approach (see Appendix Sect. B.3.2) and then used to disambiguate or to translate (depending on the error measure being reported) a test corpus. The training is performed by initializing the parameters using the method of (Kupiec 1992; Manning and Schütze 1999, p. 358) and reestimating the model using the Baum-Welch algorithm. When reestimating the HMM parameters, the log-likelihood of the training corpus is calculated after each iteration; the iterative reestimation process is finished when the difference between the log-likelihood of the last iteration and the previous one is below a certain threshold. For this setup, and only in the case of Spanish and French, we used training corpora (from the same newspapers) with around 10 million untagged words.

Supervised: The HMM-based PoS tagger is trained via MLE (see Appendix Sect. B.3.1) from a hand-tagged corpus that is independent from the one used to evaluate the PoS tagging performance, and then used to disambiguate or to translate a test corpus. Results using this setup are only provided for Spanish, as no tagged corpora are available for the other two languages. The Spanish hand-tagged corpus used comes from the Spanish newspaper *20 Minutos* and contains 21,726 words.

TLM-best: Instead of using an HMM-based SL PoS tagger, a TL model M_{TL} is used at *translation time* to select always the most likely translation into the TL. To that end, all possible disambiguation paths of each text segment are translated into the TL and scored against the TL model M_{TL}. Note that this MT setup is unfeasible for real applications, such as online MT, because the number of disambiguation paths per segment, and consequently the number of translations to perform, grows exponentially with segment length.[13]

The results achieved by the Baum-Welch MT setup may be considered as the baseline to improve upon (both are unsupervised); in contrast, the results achieved by the TLM-best setup may be considered as an approximate indication of the best results that our method could achieve, as our method transfers information about TL trigrams to an SL first-order HMM (bigrams), possibly involving some loss in accuracy.

[13] Note that a dynamic-programming approach could not be applied to reduce the computational complexity of the TLM-best setup because in most rule-based MT systems transfer cannot be given an analytical, synchronous characterization, as different transfer rules spanning 'phrases' of different length are applied to each disambiguation.

5.5 Use of a complete structural transfer MT system

In this section we study the translation performance into Catalan after training the PoS taggers for Spanish, French and Occitan by using a complete structural transfer MT system in the training phase; moreover, for Spanish we also study the PoS tagging performance. In Sect. 5.6, we study the use of a null structural transfer module to train the same HMM-based PoS taggers in order to assess the importance of having completed the development of the structural transfer module of the MT system before training.

5.5.1 Text segmentation

An adequate strategy for SL text segmentation is necessary as described in Sect. 4. The strategy followed in this section consists of segmenting at unambiguous words whose PoS tag is not present in any structural transfer pattern, or at unambiguous words appearing in patterns that cannot be matched in the lexical context in which they appear. To do so, for every pattern involving an unambiguous word, we look at the surrounding words that could be matched in the same pattern, and segmentation is performed only if none of these words have a PoS tag causing the transfer pattern to be matched. For example, to determine if an unambiguous word with the PoS tag "noun" is a segmentation point, all transfer patterns for the corresponding language pair are examined. Suppose that the tag "noun" only appears in these two structural transfer patterns: "article–noun" and "article–noun–adjective". The segmentation will be performed only if the previous and the next word cannot be assigned the "article" and "adjective" PoS tags, respectively.

In addition, an exception is taken into account; no segmentation is performed at words which start a multi-word unit whose translation could be contracted into a single word by the post-generator (for example in Spanish, *de* followed by *los*, which usually translates as *dels* (=*de*+*els*) into Catalan). Unknown words are also treated as segmentation points even though they are ambiguous, since the *lexical transfer* component has no bilingual information for them and no *structural transfer* pattern is activated at all.

5.5.2 Results

Figure 4 shows, for each language pair, the translation performance achieved by each MT setup used as a reference (see Sect. 5.4) and that attained by our new MT-oriented method after training with one training corpus randomly chosen from the set of corpora used to test the MT-oriented method (remember that we used five corpora for Spanish and French); the remaining training corpora show similar results and consequently are not discussed further in this paper. Translation performances are reported with their respective 95% (longer intervals) and 85% confidence intervals computed for the corresponding test corpus by repeatedly calculating the corresponding evaluation measure from a test corpus drawn randomly with replacement from the original one (see Sect. 5.2.2) 1,000 times. The confidence intervals reported show the range of values within the corresponding measure lying with probability 0.95 or 0.85 (depending on which confidence interval we pay attention to) for test sets with the

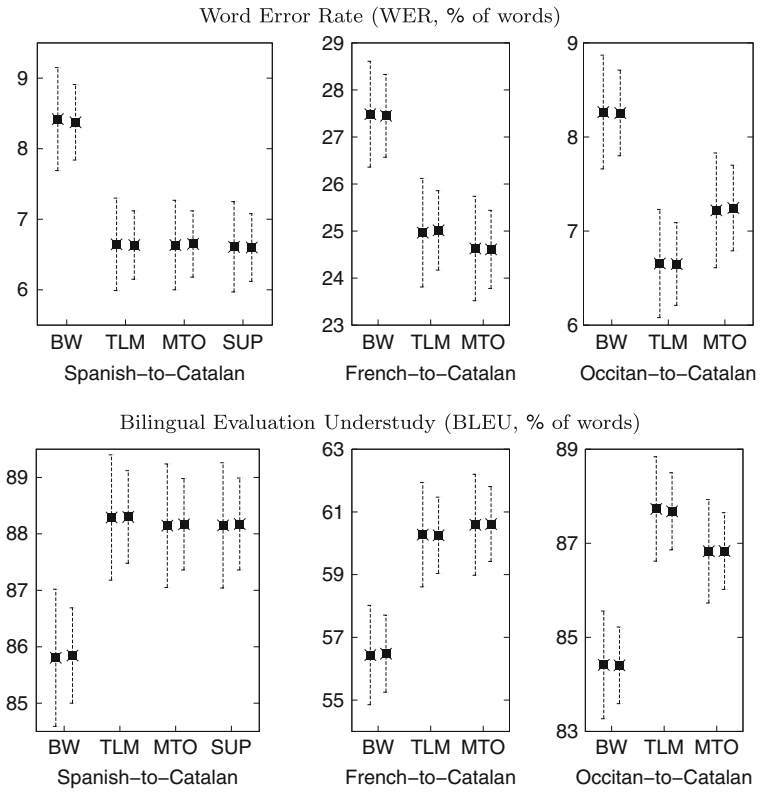

Fig. 4 WERs (top) and BLEU scores (bottom), with their respective 95% (longer intervals) and 85% confidence intervals obtained for translation from Spanish, French and Occitan into Catalan. BW stands for the results achieved by the Baum-Welch MT setup, TLM for the results achieved by the TLM-best setup, MTO for the results achieved by the MT system embedding a PoS tagger trained through our new MT-oriented training method, and SUP for the results achieved by the MT system embedding a PoS tagger trained in a supervised way (only for Spanish)

number of sentences reported in Table 3 for each evaluation corpus. Note that for a given language and confidence value, the intervals for all the MT setups are of similar width.

As can be seen in Fig. 4, the translation quality achieved by the MT-oriented training method is clearly better than is the case when the standard unsupervised approach to train HMM-based PoS taggers (the Baum-Welch algorithm), is used, and comparable (confidence intervals show a large overlapping) to the results achieved when using a TL model at translation time to score each possible translation and selecting the most likely one (TLM-best). Recall that while the results provided by the Baum-Welch setup may be considered as the baseline to improve upon, the TLM-best setup provides an approximate indication of the best results that our MT-oriented method could achieve. Note that in the case of Occitan-to-Catalan translation, the 95% confidence interval provided for the WER and the Baum-Welch MT setup overlaps with that of the MT-oriented approach; however, they do not overlap in the case of the 85% confidence interval.

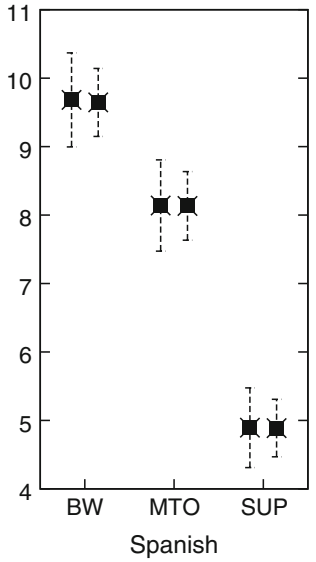

Fig. 5 Part-of-speech (PoS) tagging error rate, with respective 95% (longer intervals) and 85% confidence intervals, over all words for the Spanish PoS tagger when trained by means of the Baum-Welch algorithm (BW) using an untagged corpus, via the new MT-oriented method (MTO, also from untagged corpora), and via the MLE method from a hand-tagged corpus (SUP, supervised training)

Figure 4 also shows, but only in the case of Spanish, that the results achieved when embedding a PoS tagger trained in a supervised way are very similar to those achieved when embedding a PoS tagger trained via our unsupervised MT-oriented method; their confidence intervals show a large overlap.

Figure 5 shows the PoS tagging error rate over all words (including unknown words), together with confidence intervals when training the Spanish PoS tagger via the (unsupervised) Baum-Welch algorithm, the MT-oriented method and the supervised MLE method. Note that the performance of our (MT-oriented) approach, in terms of PoS tagging accuracy, is better than the performance achieved when training via the Baum-Welch algorithm, but it goes about one third of the way toward reaching the tagging performance achieved when training in a supervised way from hand-tagged corpora. Nevertheless, as can be seen in Fig. 4, the translation quality achieved by the MT-oriented method is almost equal to that achieved by the supervised training method.

The fact that our method achieves a translation quality comparable to that achieved by the supervised method, while PoS tagging performance is worse, may be explained by the free-ride phenomenon (very common in the case of related language pairs such as Spanish–Catalan, see Sect. 2). As PoS tags involved in a free-ride phenomenon produce the same translation, the method cannot distinguish among those tags while training (recall that the language model is based on surface forms) and the resulting tagger does not correctly tag some words, but their translations are still correct.

Finally, Fig. 6 shows the evolution of the mean and the standard deviation of the WERs for the 5 disjoint corpora used to train the Spanish PoS tagger; BLEU scores behave in a similar manner. While training, the HMM parameters were estimated and

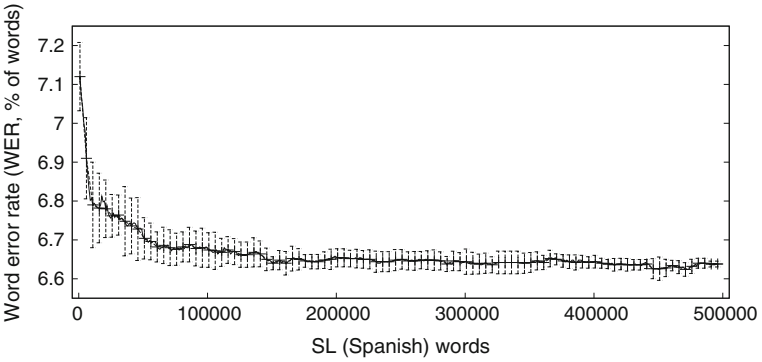

Fig. 6 Evolution of the mean and standard deviation of the WER for the five disjoint corpora used to train the Spanish PoS tagger, Catalan being the TL

the resulting performance was recorded after every 1,000 words in order to obtain these figures.

As Fig. 6 illustrates, the MT-oriented method does not need too much text to converge and it behaves in a similar way with all the training corpora used, as indicated by the standard deviation reported. Note that the other two languages require similar amounts of text to converge.

5.6 Use of a null structural transfer MT system

In the experiments reported in the previous section a full structural transfer MT system was used. Because of this, information about transfer patterns had to be taken into account when segmenting in order to make each segment independent of the adjacent ones. In this section we present a set of experiments conducted by reducing the structural transfer of the corresponding language pair to a bare minimum, i.e. context-free word-for-word translation with no structural transfer. In other words, in this section training is performed using an MT system in which the structural transfer module has no transfer patterns and, consequently, processes the input word for word without taking context into account.

5.6.1 Text segmentation

As transfer patterns have been removed from the structural transfer module it can be said that each word is treated independently from the adjacent ones after the PoS tagger.[14] This makes it possible for the method to just segment at every unambiguous word, which makes segments much smaller and reduces the number of translations to perform for each segment. As in the other experiments, unknown words are also treated as segmentation points despite being ambiguous because no bilingual information is available for them, and therefore unknown words are not translated.

[14] The orthographical operations (contractions and apostrophes) performed by the morphological generator have also been removed.

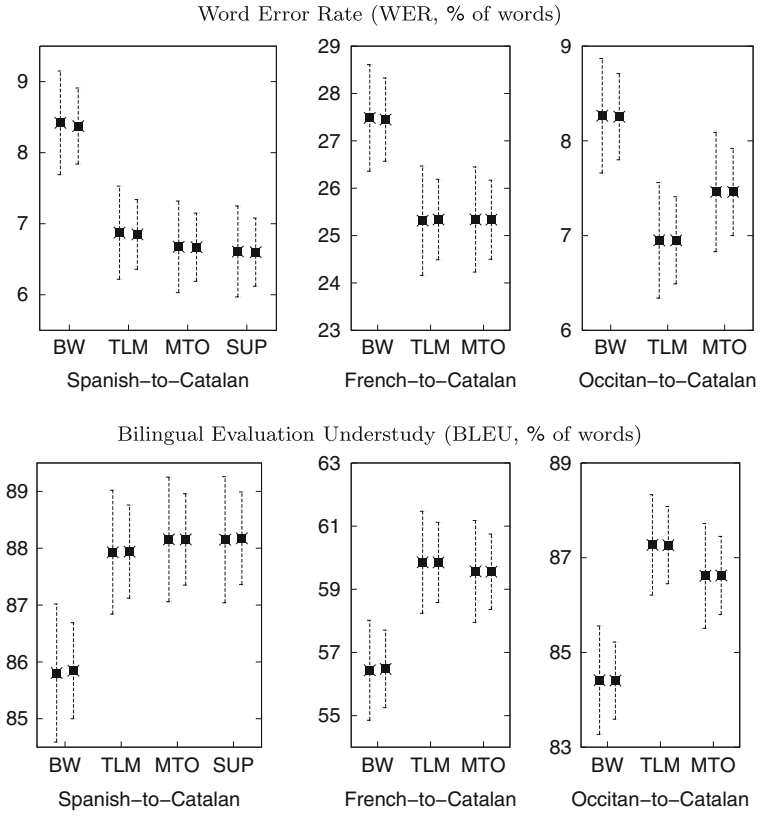

Fig. 7 WERs (top) and BLEU scores (bottom), with their respective confidence intervals, obtained for translation from Spanish, French and Occitan into Catalan. The MT-oriented method was applied by using a *null structural transfer* component while training. The TLM-best result reported was calculated by using a null structural transfer module when selecting the disambiguation path that produces the more likely translation, but the full one when performing the evaluated translation (as in the MT-oriented method). As with Fig. 4 in which a full structural transfer system was used in the training phase, BW stands for the results achieved by the Baum-Welch MT setup, TLM for the results achieved by the TLM-best setup, MTO for the results achieved by the MT-oriented method and SUP for the results achieved by supervised training method

5.6.2 Results

Figure 7 shows, for each language pair, the translation performance achieved by each MT setup and by the MT-oriented method after training the corresponding PoS tagger using a null structural transfer MT system. Note that a null structural transfer component is used during training, but the full one is used for the evaluation of translation performance. As in the experiments reported in the previous section, the Baum-Welch, supervised (only for Spanish), and TLM-best results are displayed for reference. For direct comparison with our method, in this case the TLM-best result was calculated by using a null structural transfer component when selecting the disambiguation path that produces the more likely translation, but using the full transfer system when performing the evaluated translation.

Comparing Fig. 7 with Fig. 4 in which a full structural transfer MT system is used by the training algorithm, the results are quite similar in the case of Spanish–Catalan even though in this last experiment no actions are performed in order to solve the grammatical divergences between the SL and the TL during training. However, in the case of French–Catalan and Occitan–Catalan, the PoS taggers trained by using a null structural transfer MT system are worse than those obtained by using the full structural transfer MT system. More precisely, for French the WER is around 0.7% worse (compare the centre of the confidence intervals), whereas for Occitan the WER is only around 0.2% worse, with BLEU scores showing the same behaviour. Note that, although these results are slightly worse than those reported when a full structural transfer MT system is used for training, the resulting PoS taggers are still better than those trained via the Baum-Welch algorithm. Note that, in the case of Occitan and very slightly in the case of French, the 95% confidence intervals for the Baum-Welch and the MT-oriented methods overlap. However, the 85% confidence intervals for BLEU do not overlap, i.e. with probability 0.85 the performance of the MT-oriented training method is always better (according to BLEU) than that of the Baum-Welch algorithm for test sets of the size of those used in the evaluation (see the number of sentences of the evaluation corpora on Table 3).

The fact that the results achieved when using a null structural transfer system, compared with the results achieved when using a full structural one, are different for different language pairs gives an idea as to how related two language pairs are. Note that when no transfer rules are taken into account, no actions are performed to treat the grammatical divergences between the languages involved. From this point of view it may be said that Spanish and Catalan are more related than Occitan and Catalan, or French and Catalan.

With respect to the PoS tagging performance of the Spanish PoS tagger, it is of the same order as the one reported in Fig. 5. Finally, it is worth mentioning that the amount of text required for convergence when using a null structural transfer MT system while training is the same as the amount of text required for convergence when using a full structural transfer MT system in the training phase.

5.7 Time complexity reduction by pruning disambiguation paths

We repeated the same experiments reported in Sect. 5.5, in which full structural transfer was used, with the same corpora, but applying the pruning method described in Sect. 3 as follows: First, the initial model was computed using the method of (Kupiec 1992), i.e. the same unsupervised initialization method used when training via the Baum-Welch algorithm. After that, the HMM-based PoS tagger was trained by using information from the TL as described in Sect. 2 in conjunction with the pruning technique introduced in Sect. 3. The HMM used for pruning was updated every 1,000 words as explained in Sect. 3.3. To this end, the weight function $\varphi(x)$ used in Eq. (18) was chosen to grow linearly from 0 to 1 with the fraction x of the SL corpus processed so far, as in (19):

$$\varphi(x) = x. \tag{19}$$

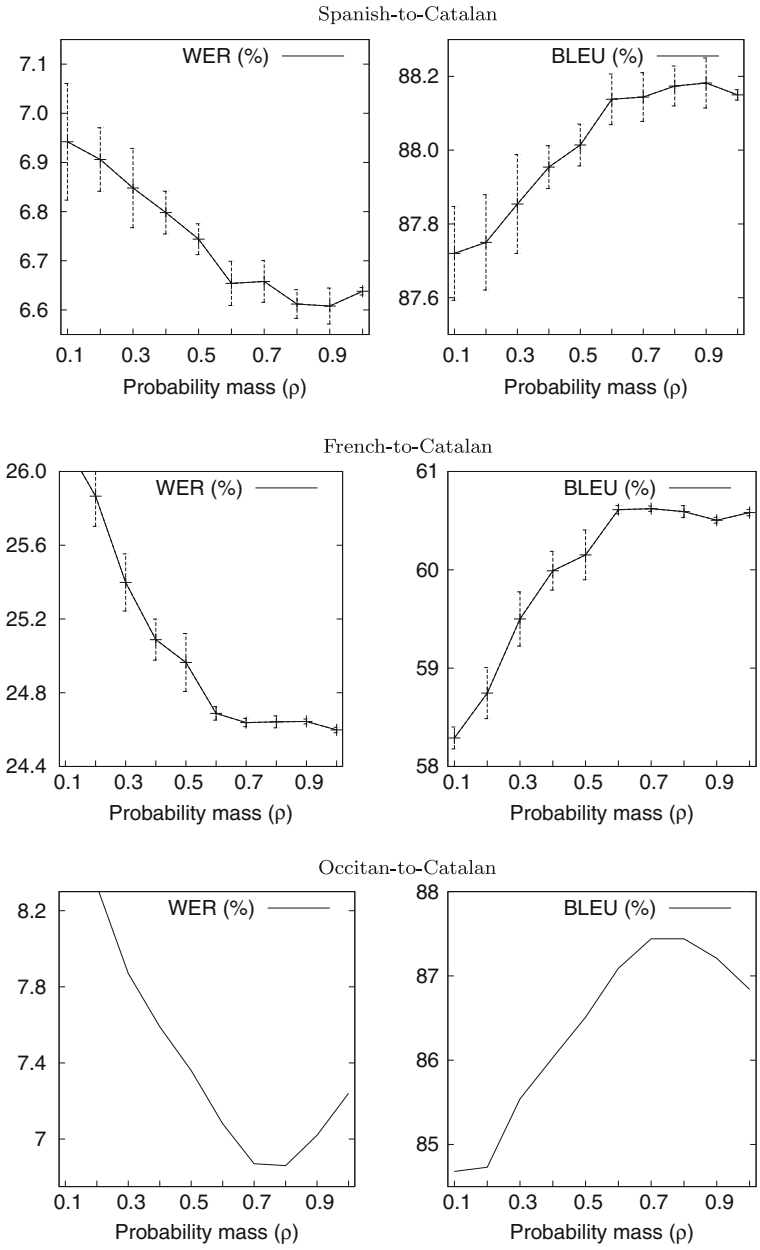

Fig. 8 For the different values of ρ used, mean and standard deviation of the WERs (left) and of the BLEU scores (right) achieved after training the Spanish and French PoS taggers with the different training corpora used, and WERs and BLEU scores achieved after training the Occitan PoS tagger

In order to determine the appropriate mass probability threshold ρ that speeds up our training method without degrading its performance, we considered a set of values for ρ between 0.1 and 1.0 at increments of 0.1. Note that when $\rho = 1.0$, no pruning

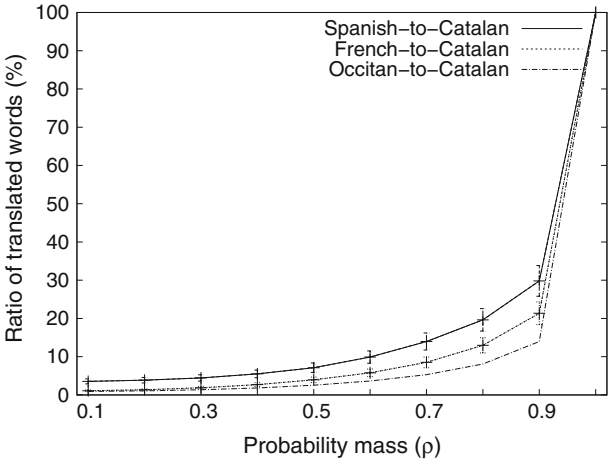

Fig. 9 For each language, the percentage of translated words for each value of the probability mass threshold ρ. The percentage of translated words is calculated over the total number of words that are translated when no pruning is done

is done, i.e. all possible disambiguation paths for each segment are translated into the TL.

Figure 8 shows the mean and standard deviation of the WER and the BLEU score, respectively, achieved by the MT system embedding the Spanish and the French PoS taggers for the different values of ρ after training with the different corpora used. The WER and the BLEU score achieved by the Occitan–Catalan MT system embedding the Occitan PoS tagger after training are also given for all tested values of ρ. As can be seen, the three languages behave in a similar way, the best results being achieved for values of ρ between 0.7 and 0.9. Note that WERs and BLEU scores achieved are indeed better for Spanish and Occitan than those achieved when no pruning is performed. This may be explained by the fact that the fractional counts associated to discarded disambiguation paths are assumed to be null; however, when no pruning is performed these fractional counts are small, but never null. Finally, note that in the case of Spanish, the standard deviation is smaller when no pruning is done ($\rho = 1.0$).

As to how many translations are avoided with the proposed pruning method, Fig. 9 shows, for the three languages being studied, the average ratio and standard deviation of the number of words finally translated with respect to the total number of words translated when no pruning is performed. As can be seen, for the value of ρ that produces the most accurate PoS tagger to be used in MT for each language (0.9 for Spanish, 0.7 for French, and 0.8 for Occitan), the percentage of words translated is around 25%. This percentage can be seen as roughly proportional to the percentage of disambiguation paths needed to reach the corresponding mass probability threshold.

6 Discussion

In this paper we have described and tested a new (unsupervised) method to train PoS taggers to be used in MT. This new method simplifies the process of building an RBMT

system from scratch as no hand-tagged corpora are needed to obtain better results than the standard unsupervised Baum-Welch EM algorithm. MT system developers using this new method only need to build the rest of modules of the translation engine before training the HMM-based PoS taggers of that MT system.

Our training method uses the remaining modules of the MT system in which the resulting PoS tagger is to be embedded to generate translations that are then scored using an unsupervisedly trained TL model. Then these TL scores are used to estimate the HMM parameters of the PoS tagger. The use of TL information to train an SL tagging model opens a new line of research in the construction of SL PoS taggers, and other SL models, to be used in MT.

We tested our method on three different languages (Spanish, French and Occitan), all being translated into Catalan. The performance of our approach was compared with three different MT configurations: the use of a PoS tagger trained through the standard unsupervised approach (Baum-Welch), the use of a PoS tagger trained in a supervised way from hand-tagged corpora (supervised, only for Spanish), and the use of a TL model at translation time (instead of a PoS tagger) to always select the most likely translation into the TL (TLM-best). The Baum-Welch MT setup may be considered as the baseline whose results are improved upon, while the TLM-best setup may be seen as an approximate indication of the best results that our MT-oriented training method could achieve (see Sect. 5.4).

The experiments were conducted using different training corpora, where available, so as to test whether the amount of text needed for convergence and the behaviour, in terms of performance, was the same for all of them. Furthermore, we reported the confidence intervals in conjunction with translation performance scores so as to better enable the reader to interpret the significance of any such differences.

For all three language pairs our method gives better results than the Baum-Welch trained PoS tagger, and results of the order of those achieved by the TLM-best setup. Note that, although our training algorithm also uses a TL model to score translations, this is only done for training, never at translation time; therefore, the PoS tagger is as fast as any other HMM-based PoS tagger. However, the use of the TLM-best setup makes translation much slower, since all possible disambiguations of a given text segment must be translated and scored against a TL model before selecting the most likely translation, which makes the TLM-best setup unfeasible for some real applications such as online MT. A possible criticism here may be that the computational complexity of the TLM-best setup could be reduced to a negligible amount by using a dynamic programming (DP) approach. However, this is not the case because a DP algorithm would require transfer to be described by a synchronous analytical function, which is not feasible in most RBMT systems, including Apertium (see Footnote 13).

Furthermore, as the results on the Spanish language show, the translation quality achieved by the MT system embedding a PoS tagger trained via this new unsupervised method is comparable to that achieved by the same MT system when embedding a PoS tagger trained in a supervised manner from hand-tagged corpora. Regarding the PoS tagging accuracy, however, our method performs better than the classical unsupervised approach but worse than the supervised one. This different behaviour of PoS tagging errors and translation errors may be due to the existence of free rides (words being translated in the same way regardless of the selected PoS tag); this is because

our method cannot distinguish between PoS tags leading to the same translation. Therefore, it can be concluded that, as expected, our method is a good choice to train PoS taggers for MT, but not as good as the supervised one to train general-purpose PoS taggers to be used in other NLP applications.

For the training experiments we used two different MT systems, one having a structural transfer module that performs some operations, such as gender and number agreement or word reordering to meet the TL grammatical rules (see Sect. 5.5), and another that does not perform any structural transfer operation (see Sect. 5.6). The latter MT system can be said to process each word independently of the adjacent ones after the PoS tagger. It was shown that the results achieved in both cases are quite similar for Spanish, and less similar for Occitan and French.

The fact that the results achieved when no structural transformations are applied by the MT system used for training are quite similar to those achieved when using a full structural transfer module may be explained by the fact that closely related languages (such as Spanish and Catalan or Occitan and Catalan) have little grammatical divergence. This result indicates that in order to benefit from the training method we propose, RBMT developers do not need to wait until they have a complete MT system, as they can train a PoS tagger of a reasonable quality before developing the structural transfer module.

The main disadvantage of the method presented in this paper is that the number of translations to perform for each SL text segment grows exponentially with the segment length. In order to overcome this problem a disambiguation path pruning technique based on a priori knowledge, obtained in an unsupervised way from the SL, was proposed and tested. This pruning method is based on the assumption that any reasonable model of SL tags may prove helpful in choosing a subset of possible disambiguation paths, the correct one being included in that subset. Moreover, the model used for pruning can be updated during training with the new data collected while training. The results reported in Sect. 5.7 show that on the one hand, the pruning method described avoids more than 70% of the translations to be performed, and on the other hand, that the results achieved by our training method improve slightly for some language pairs if improbable disambiguation paths are not taken into account.

Finally, it must be mentioned that the training method presented here may benefit less-resourced languages like Occitan which lacks large electronic resources.

7 Future work

As a consequence of the free-ride phenomenon (SL words being translated into the same TL word for every possible disambiguation), more than one disambiguation path $\boldsymbol{g}$ may produce the same translation $\tau(\boldsymbol{g}, s)$. In that case, the (fractional) contribution $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s)$ of each disambiguation path $\boldsymbol{g}$ to the shared translation $\tau(\boldsymbol{g}, s)$ was approximated as being equal via Eq. (11). We plan to study the real impact of the free-ride phenomenon on translation performance achieved by the MT system which embeds the resulting PoS tagger. According to the results of this study, we hope to introduce better alternatives to Eq. (11). On the one hand, an initial model like the one used to prune unlikely disambiguation paths may be used to better estimate that factor;

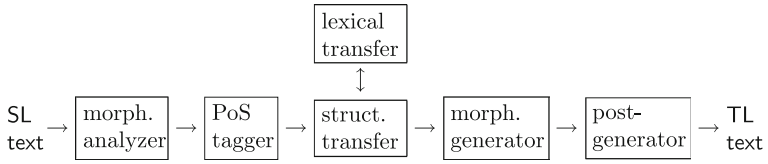

Fig. 10 Main modules of the open-source shallow-transfer MT engine Apertium used in the experiments (see Appendix A)

on the other hand, the EM algorithm may be applied to iteratively better estimate that contribution, but at the cost of increasing the overall training time.

Concerning the path pruning method described in this paper to speed up the HMM-based PoS tagger training method presented, 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 during training (an *annealing schedule*), 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 latter method 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.

Finally, we plan to devise a DP algorithm to reduce the time complexity of null-transfer training when MT involves closely related languages. This can be done because when a null structural transfer MT system is used, each word is processed after the PoS tagger independently of the adjacent ones, allowing the translation model M_{trans} to be described by an analytical function.

Appendix

A. The Apertium machine translation platform

This appendix describes the open-source shallow-transfer MT engine Apertium[15] (Armentano-Oller et al. 2006) used for the experiments. Apertium follows the shallow transfer approach shown in Fig. 10:

- A *morphological analyzer* which tokenizes the text in surface forms and delivers, for each surface form, one or more *lexical forms* consisting of *lemma*, *lexical category* and morphological inflection information.
- A *part-of-speech tagger* (categorial disambiguator) which chooses, using a first-order hidden Markov model (Baum and Petrie 1966; Cutting et al. 1992) (HMM), one of the lexical forms corresponding to an ambiguous surface form. This is the module trained in the experiments by using the remaining modules of the MT engine; therefore, this module is not used by the training algorithm.
- A *lexical transfer* module which reads each SL lexical form and delivers the corresponding TL lexical form by looking it up in a bilingual dictionary.

[15] The MT engine, documentation, and linguistic data for different language pairs can be downloaded from http://apertium.sf.net.

– A *structural transfer* module (parallel to the lexical transfer component) which uses a finite-state chunker to detect patterns (such as "article–noun–adjective") of lexical forms which need to be processed for word reordering, agreement, etc., and then performs those operations.
– A *morphological generator* which delivers a TL surface form for each TL lexical form, by suitably inflecting it.
– A *post-generator* which performs orthographic operations such as contractions (e.g. Spanish *del* = *de*+*el*) and apostrophes (e.g. Catalan *l'institut* = *el*+*institut*).

The Apertium MT engine is completely independent of the linguistic data used to translate for a given language pair. Linguistic data is coded using XML-based formats;[16] this allows for easy data transformation and maintenance. For the experiments we used linguistic data for three different languages pairs; in particular, we used the language-pair packages `apertium-es-ca-1.0.2`, `apertium-fr-ca-0.9`, and `apertium-oc-ca-1.0.2` to test our approach for the Spanish–Catalan, French–Catalan, and Occitan–Catalan (Armentano-Oller and Forcada 2006) language pairs. It must be noted that the HMM parameters for the French and the Occitan PoS taggers currently distributed with the corresponding linguistic packages were obtained following the approach presented in this paper.

B. HMM-based part-of-speech taggers

This appendix overviews the principles of HMMs and their application as PoS taggers in the field of NLP.

B.1 HMMs for part-of-speech tagging

An HMM (Baum and Petrie 1966; Rabiner 1989) is defined as $\lambda = (\Gamma, \Sigma, A, B, \pi)$, where Γ is the set of hidden 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 $\sigma \in \Sigma$ being emitted from each hidden state $\gamma \in \Gamma$, and the vector π, with dimensionality $|\Gamma|$, defines the initial probability of each hidden state. The system produces an output each time a state is reached after a transition. A deeper description of this kind of statistical model may be found in Cutting et al. (1992) and (Manning and Schütze 1999, ch. 9).

When an HMM is used to perform PoS tagging, each HMM state γ is made to correspond to a different PoS tag, 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. 1992), that is, the set of all possible PoS tags that a word could receive, but sometimes it may be useful to have finer classes, such as a word class containing only a single, very frequent, ambiguous word. In addition, unknown words (that is,

[16] The XML formats (http://www.w3.org/XML/) for each type of linguistic data are defined through conveniently designed XML document-type definitions (DTDs) which may be found inside the apertium package.

words not found in the lexicon) are usually assigned the set of *open* categories, i.e. the set of PoS tags (categories) which are likely to grow by addition of new words to the lexicon of a language: nouns, verbs, adjectives, adverbs and proper nouns. Moreover, when an HMM is used to perform PoS tagging, the estimation of the initial probability of each state can be avoided by assuming that each sentence always begins with the end-of-sentence mark. In this way $\pi(\gamma)$ is 1 when γ is the end-of-sentence mark, and 0 otherwise.

PoS ambiguities are solved by assigning to each word the PoS tag found in the PoS tag sequence that maximizes its likelihood given the sequence of observable outputs (word classes). The model assumes that the PoS tag of each word depends only on the previous word when a first-order HMM is used, or on the n previous words when an n-th order HMM is considered.

Once the HMM parameters have been estimated (independently of the method used for training), the Viterbi algorithm (Manning and Schütze 1999, p. 332) is used for disambiguation. This DP algorithm efficiently computes the sequence of PoS tags that maximizes its likelihood given the observable outputs. The algorithm can be applied to text segments smaller than whole sentences, provided that they are delimited by a sequence of n unambiguous words, n being the order of the HMM. This can safely be done because unambiguous words 'unhide' or reveal the hidden state of the HMM (Cutting et al. 1992, Sect. 3.4), making the disambiguation of the words following that sequence of n unambiguous words independent of the ambiguous ones preceding it.

B.2 Parameter smoothing

Independently of the method used to estimate the HMM parameters, a smoothing technique should be used in order to avoid null probabilities for those state-to-state transitions and output emissions that have not been seen in the training corpus. Here we describe the smoothing applied in the experiments reported in Sect. 5.

Parameter smoothing can be conveniently achieved using a form of *deleted interpolation* (Jelinek 1997, ch. 4) in which weighted estimates are taken from first-order models and a uniform probability distribution.[17]

B.2.1 State-to-state transition probabilities

The smoothing of the state-to-state transition probabilities consists of a linear combination of bigram and unigram probabilities, as in (20):

$$a_{\gamma_i\gamma_j} = P(\gamma_j|\gamma_i) = \lambda(\gamma_i)\frac{n(\gamma_i\gamma_j)}{\sum_{\gamma_k\in\Gamma} n(\gamma_i\gamma_k)} + (1-\lambda(\gamma_i))P(\gamma_j) \tag{20}$$

Here $\lambda(\gamma_i)$ is the smoothing coefficient for tag bigrams, $n(\gamma_i\gamma_j)$ is the count of the number of times tag γ_i is followed by tag γ_j in the training corpus, and $P(\gamma_j)$ is the probability of having seen the tag γ_j.

[17] The equations provided here can be easily extended to smooth the parameters of a higher-order HMM.

Jelinek (1997) computes the values of the smoothing coefficients by splitting the training corpus into the *kept* part—the larger one from where frequency counts are collected—and the *held-out* part, used to collect more counts and estimate the value of the smoothing coefficients.

A simple, approximate way to estimate the value of the smoothing coefficients without having to deal with a held-out corpus is the *successive linear abstraction* method proposed by Brants and Samuelsson (1995)[18] and used in our experiments, as in (21):

$$\lambda(\gamma_i) = \frac{\sqrt{n(\gamma_i)}}{1 + \sqrt{n(\gamma_i)}} \tag{21}$$

Here $n(\gamma_i)$ is the number of occurrences of the tag γ_i in the training corpus.

Nevertheless, in spite of the smoothing techniques used, when a tag bigram ends at a previously unseen tag γ_j, the final probability is still zero because the unigram probability $P(\gamma_j)$ is null. To avoid this problem, unigram probabilities are also smoothed via Eq. (22):

$$P(\gamma_j) = \mu \frac{n(\gamma_j)}{\sum_{\gamma_k \in \Gamma} n(\gamma_k)} + (1 - \mu) \frac{1}{|\Gamma|} \tag{22}$$

Here the second term estimates, in the absence of further information, the probability of each tag as being equally likely.[19] The weight of this second term in the final smoothed probability $P(\gamma_j)$ depends on the smoothing coefficient μ, which is made to depend on the length L of the training corpus, and calculated in an analogous way to that proposed by Brants and Samuelsson (1995), as in (23):

$$\mu = \frac{\sqrt{L}}{1 + \sqrt{L}} \tag{23}$$

B.2.2 Emission probabilities

Although observable outputs are made to correspond to word classes (see Sect. B.1), which reduces the total number of observable outputs and the data sparseness problem, emission probabilities still need to be smoothed.

The smoothing of the emission probabilities is done in an analogous way to that used to smooth the state-to-state transition probabilities, as in (24):

$$b_{\gamma_j}(\sigma_k) = P(\sigma_k|\gamma_j) = \lambda(\gamma_j) \frac{n(\sigma_k, \gamma_j)}{\sum_{\sigma':\gamma_j \in \sigma'} n(\sigma', \gamma_j)} + (1 - \lambda(\gamma_j)) P_{\gamma_j}(\sigma_k) \tag{24}$$

[18] Note that any reasonable smoothing technique could have been used instead; this is not central to the method we present.

[19] This can be done quite safely since the total number of tags is known and equal to $|\Gamma|$.

Here $\lambda(\gamma_j)$ is the smoothing coefficient calculated as shown in Eq. (21), $n(\sigma_k, \gamma_j)$ is the count of the number of times word class σ_k is emitted from tag γ_j, and $P_{\gamma_j}(\sigma_k)$ is the probability of word class σ_k taking into account only those ambiguity classes which can be effectively emitted from tag γ_j, as in (25):

$$P_{\gamma_j}(\sigma_k) = \begin{cases} \frac{P(\sigma_k)}{\sum_{\sigma':\gamma_j \in \sigma'} P(\sigma')} & \text{if } \gamma_j \in \sigma_k \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

Here $P(\sigma_k)$ is the (smoothed) probability of ambiguity class σ_k. This probability is smoothed in an analogous way to that used for $P(\gamma_j)$ (see Eq. (22)), as in (26):

$$P(\sigma_k) = \mu \frac{n(\sigma_k)}{\sum_{\sigma_l \in \Sigma} n(\sigma_l)} + (1 - \mu) \frac{1}{|\Sigma|} \tag{26}$$

Here μ refers to the smoothing coefficient calculated as shown in Eq. (23), and $n(\sigma_k)$ is the count of the number of times ambiguity class σ_k appears in the training corpus. As in Eq. 22, in the absence of further information all ambiguity class are assumed to be equally likely (second term).

Equation 24 does not directly use the probability $P(\sigma_k)$, because the use of $P(\sigma_k)$ would cause the probability $P(\sigma_k|\gamma_j)$ to be non-null also in those cases in which $\gamma_j \notin \sigma_k$, i.e. in those cases in which σ_k cannot be emitted from tag γ_j.

B.3 General-purpose HMM training methods

In this section we review the classical supervised and unsupervised methods used to train general-purpose HMM-based PoS taggers.

B.3.1 The maximum-likelihood estimate method

When a hand-tagged corpus is available the HMM parameters can be estimated directly via the maximum-likelihood estimate (MLE) method in a supervised manner in conjunction with an smoothing technique (see Appendix Sect. B.2). To apply any smoothing technique, frequency counts $n(\cdot)$ must be collected. Since in a tagged corpus each segment has only one possible disambiguation, it is easy to collect these frequency counts and to use them to estimate the transition and the emission probabilities via Eqs. (20) and (24), respectively.

B.3.2 The Baum-Welch expectation-maximization method

When hand-tagged corpora are not available an unsupervised method must be applied. The classical unsupervised method to train HMMs is the forward-backward algorithm, also known as the Baum-Welch algorithm (Baum 1972; Cutting et al. 1992).

The Baum-Welch algorithm is a special case of the *expectation-maximization* method. This training algorithm works as follows; as the model is unknown, the probability of the observation sequence can be worked out with an initial model that may be

randomly chosen, or estimated from corpora via the method of (Kupiec 1992; Manning and Schütze 1999, p. 358), or indeed via any other reasonable initialization method. Once an initial model is chosen, the method works by giving the highest probability to the state transitions and output emissions used the most. In this way a revised, more accurate model is obtained. This model can in turn be reestimated using the same procedure iteratively. After each Baum-Welch iteration, the new HMM parameters may be shown to give a higher probability to the observation sequence (Baum 1972). A deeper and more formal description of the Baum-Welch algorithm is given in Cutting et al. (1992) and Manning and Schütze (1999, ch. 9).

In the experiments reported in Sect. 5 the Baum-Welch algorithm is used in conjunction with the smoothing techniques described in Appendix Sect. B.2, as for the MLE training method and the MT-oriented training method introduced next.

B.4 Tagset definition

The tagset to be used by a PoS tagger must be carefully designed. The main goal when defining the tagset is to use the smallest possible number of tags, grouping finer tags into coarse ones, but avoiding grouping tags having different syntactic roles. Notice that finer tags convey more information, but at the same time they increase considerably the number of HMM parameters to be estimated, worsening the problem of data sparseness, and thus increasing the number of parameters that achieve a null-frequency count. In particular, when PoS taggers are used in MT systems, what really counts is to be able to distinguish analyses leading to different translations.

Sometimes, in order to improve accuracy, partially lexicalized HMMs may be useful. In partially lexicalized HMMs some word classes are chosen to hold only a single word, and, therefore, they are finer than ambiguity classes (Cutting et al. 1992). In this way the model can deal better with the peculiarities of certain words. The lexicalization described by Cutting et al. (1992) only enriches the lexical model, because only new observables (single-word classes) are defined. Kim et al. (1999) and Pla and Molina (2004) describe a different lexicalization technique that also adds new states for those tags assigned to words receiving a specific treatment; in this case the syntactic model is also modified because new PoS tags are added. In our experiments we used this last kind of lexicalized HMMs.

Acknowledgements Work funded by the Spanish Ministry of Science and Technology through project TIC2003-08601-C02-01 and by the Spanish Ministry of Education and Science and the European Social Fund through research grant BES-2004-4711 and project TIN2006-15071-C03-01. We thank Rafael C. Carrasco (Universitat d'Alacant, Spain) for very useful comments on this work. We also thank Geoffrey Sampson (University of Sussex, England) for his Simple Good-Turing implementation.

References

Armentano-Oller C, Carrasco RC, Corbí-Bellot AM, Forcada ML, Ginestí-Rosell M, Ortiz-Rojas S, Pérez-Ortiz JA, Ramírez-Sánchez G, Sánchez-Martínez F, Scalco MA (2006) 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, vol 3960 of lecture notes in computer science. Itatiaia, RJ, Brazil: Springer-Verlag, pp 50–59

Armentano-Oller C, Forcada ML (2006) Open-source machine translation between small languages: Catalan and Aranese Occitan. In: Proceedings of strategies for developing machine translation for minority languages (5th workshop on speech and language technology for minority languages), Genoa, Italy, pp 51–54

Arnold D (2003) Why translation is difficult for computers. In: Somers H (ed) Computers and translation: a translator's guide. Amsterdam/Philadelphia: John Benjamins, pp 119–142

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

Baum LE, Petrie T (1966) Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat 37(6):1554–1563

Brants T, Samuelsson C (1995) Tagging the Teleman corpus. In: Proceedings of the 10th Nordic conference of computational linguistics, Helsinki, Finland, pp 7–20

Brill E (1992) A simple rule-based part-of-speech tagger. In: Proceedings of the 3rd applied natural language processing conference, Trento, Italy, pp 152–155

Brill E (1995a) Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. Comput Linguist 21(4):543–565

Brill E (1995b) Unsupervised learning of disambiguation rules for part of speech tagging. In: Proceedings of the third workshop on very large corpora, Somerset, NJ, pp 1–13

Brown PF, Della Pietra SA, Della Pietra VJ, Mercer RL (1993) The mathematics of statistical machine translation: parameter estimation. Comput Linguist 19(2):263–311

Carbonell J, Klein S, Miller D, Steinbaum M, Grassiany T, Frei J (2006) Context-based machine translation. In: Proceedings of the 7th conference of the association for machine translation in the Americas. Visions for the future of machine translation, Cambridge, MA, pp 19–28

Carl M, Way A (eds) (2003) Recent advances in example-based machine translation, vol 21. Dordrecht/Boston/London: Kluwer Academic Publishers.

Cutting D, Kupiec J, Pedersen J, Sibun P (1992) A practical part-of-speech tagger. In: Proceedings of the 3rd applied natural language processing conference, Trento, Italy, pp 133–140

Dermatas E, Kokkinakis G (1995) Automatic stochastic tagging of natural language texts. Comput Linguist 21(2):137–163

Dien D, Kiem H (2003) POS-tagger for English-Vietnamese bilingual corpus. In: Proceedings of the workshop on building and using parallel texts: data driven machine translation and beyond, at the human language technology and the north American chapter of the association for computational linguistics joint conference, Edmonton, Canada, pp 88–95

Efron B, Tibshirani RJ (1993) An introduction to the bootstrap Vol. 57 of monographs on statistics and applied probability. London, UK: Chapman & Hall/CRC

Foster G, Isabelle P, Plamondon P (1997) Target text mediated interactive machine translation. Mach Transl 2(1–2):175–194

Gale WA, Church KW (1990) Poor estimates of context are worse than none. In: Proceedings of the third DARPA workshop on speech and natural language. San Mateo, CA: Morgan Kaufmann Publishers Inc., pp 283–287

Gale WA, Sampson G (1995) Good-turing frequency estimation without tears. J Quant Linguist 2(3): 217–237

Jelinek F (1997) Statistical methods for speech recognition. Cambridge, MA: MIT Press

Kim JD, Lee SZ, Rim HC (1999) HMM specialization with selective lexicalization. In: Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora, College Park, MD, pp 121–127

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the conference on empirical methods in natural language processing. Barcelona, Spain, pp 388–395

Koehn P (2008) Statistical machine translation. Cambridge, UK: Cambridge University Press

Kupiec J (1992) Robust part-of-speech tagging using a hidden Markov model. Comput Speech Lang 6(3):225–242

Levenshtein VI (1965) Binary codes capable of correcting deletions, insertions, and reversals. Doklady Akademii Nauk SSSR 163(4):845–848. English translation in Soviet Physics Doklady 10(8):707–710 (1966)

Manning CD, Schütze H (1999) Foundations of statistical natural language processing. Cambridge, MA: MIT Press

Merialdo B (1994) Tagging English text with a probabilistic model. Comput Linguist 20(2):155–171

Nagao M (1984) Framework of a mechanical translation between Japanese and English by analogy principle. In: Elithorn A, Banerji R (eds) Artificial and human intelligence. Amsterdam, The Netherlands: North Holland, pp 173–180

Och FJ (2005) Statistical machine translation: foundations and recent advances. Tutorial at MT Summit X, Phuket, Thailand

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: 40th Annual meeting of the association for computational linguistics. Association for Computational Linguistics, Philadelphia, PA, pp 311–318

Pla F, Molina A (2004) Improving part-of-speech tagging using lexicalized HMMs. Nat Lang Eng 10(2):167–189

Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc Inst Electr Electron Eng (IEEE) 77(2):257–286

Sánchez-Villamil E, Forcada ML, Carrasco RC (2004) Unsupervised training of a finite-state sliding-window part-of-speech tagger. In: Advances in natural language processing, proceedings of the 4th international conference EsTAL (España for Natural Language Processing), Vol 3230 of lecture notes in computer science. Alicante, Spain: Springer-Verlag, pp 454–463

Sánchez-Martínez F, Pérez-Ortiz JA, Forcada ML (2004a) Cooperative unsupervised training of the part-of-speech taggers in a bidirectional machine translation system. In: Proceedings of the tenth conference on theoretical and methodological issues in machine translation, Baltimore, MD, pp 135–144

Sánchez-Martínez F, Pérez-Ortiz JA, Forcada ML (2004b) 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 the 4th international conference EsTAL (España for Natural Language Processing), vol 3230 of lecture notes in computer science. Alicante, Spain: Springer-Verlag, pp 137–148

Sánchez-Martínez F, Pérez-Ortiz JA, Forcada ML (2006) Speeding up target-language driven part-of-speech tagger training for machine translation. In: Advances in artificial intelligence, proceedings of the 5th Mexican international conference on artificial intelligence, vol 4293 of lecture notes in computer science. Apizaco, Tlaxcala, Mexico: Springer-Verlag, pp 844–854

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J (2006) A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th conference of the association for machine translation in the Americas. Visions for the future of machine translation, Cambridge, MA, pp 223–231

Stolcke A (2002) SRILM—an extensible language modeling toolkit. In: Proceedings of the international conference on spoken language processing, Denver, CO, pp 901–904

Yarowsky D, Ngai G (2001) Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In: Proceedings of the second meeting of the North American chapter of the association for computational linguistics, Pittsburgh, PA, pp 200–207