A Phrase-Based Hidden Semi-Markov Approach to Machine Translation

Jesús Andrés-Ferrer
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
jandres@iti.upv.es

Alfons Juan
Dpto. de sist. informáticos y computación
Universidad Politécnica de Valencia
ajuan@dsic.upv.es

Abstract

Statistically estimated phrase-based models promised to further the state-of-the-art, however, several works reported a performance decrease with respect to heuristically estimated phrase-based models. In this work we present a latent variable phrase-based translation model inspired by the hidden semi-Markov models, that does not degrade the system. Experimental results report an improvement over the baseline. Additionally, it is observed that both Baum-Welch and Viterbi trainings obtain the very same result, suggesting that most of the probability mass is gathered into one single bilingual segmentation.

1 Introduction

The machine translation problem is stated as the problem of translating a *source* sentence, $\boldsymbol{x}_1^J$, into a *target* sentence, $\boldsymbol{y}_1^I$. In accordance with the statistical approach to machine translation, the optimal translation $\hat{\boldsymbol{y}}$ of a source sentence $\boldsymbol{x}$ is given by the fundamental equation of statistical machine translation (Brown and others, 1993)

$$\hat{\boldsymbol{y}} = \operatorname*{arg\,max}_{\boldsymbol{y} \in \mathcal{Y}^\star} p(\boldsymbol{x} \mid \boldsymbol{y})\, p(\boldsymbol{y}) \tag{1}$$

where $p(\boldsymbol{x} \mid \boldsymbol{y})$ is approximated by an *inverse translation model* and *p(y)* is modelled with a *language model*; which is usually instanced by a *n-gram language model* (Chen and Goodman, 1996).

The first approaches to model the translation probability in Eq. (1), were based on word dictionaries. These word-based models, the so-called *IBM translation models* (Brown and others, 1993), tackled the problem with word-level dictionaries plus alignments between words. However, current systems model the inverse conditional probability in Eq. (1) using *phrase dictionaries*. A phrase is understood here as any sequence of source or target words. This phrase-based methodology stores specific sequences of target words (*target phrase*) into which a sequence of source words (*source phrase*) is translated.

However, a key concept of this approach is the procedure through which these phrase pairs are inferred. The common approach consists in using the IBM alignment models (Brown and others, 1993) to obtain a symmetrised alignment matrix from which *coherent* phrases are extracted (Och and Ney, 2004). Then, a simple count normalisation is carried out in order to obtain a conditional phrase dictionary.

Alternatively, some approaches infer the phrase dictionaries statistically. For instance, a joint probability model for phrase-based estimation is proposed in (Marcu and Wong, 2002). In that work, all possible segmentations were extracted using the EM algorithm (Dempster et al., 1977), without any matrix alignment constraint, in contrast to the approach followed in (Och and Ney, 2004). Based on this work, another work (Alexandra Birch and Koehn, 2006), constrained the EM to only consider phrases which agree with the alignment matrix, thus reducing the size of the phrase dictionaries (or tables).

A possible drawback of the above phrase-based models is that they are not conditional, but joint models that require a re-normalisation post-processing in order to obtain a conditional model. However, a generative conditional phrase-based model presented in (DeNero et al., 2006) showed a worsening of phrase dictionaries.

Proceedings of the 13th Annual Conference of the EAMT, pages 168–175,
Barcelona, May 2009

In this work, we propose a conditional *phrase-based hidden semi-Markov model (PBHSMM)* that improves the phrase-dictionary estimation. Although, the improvements are not impressive, bare in mind that the main property of this model is its clear theoretical foundation, since it is based on a well-known statistical modelling technique, the so-called HSMM (Ostendorf et al., 1996). This allow us to include several statistical improvements into future revisions of the model (see section 5). A previous work (Andrés-Ferrer and Juan-Císcar, 2007) already presented a conditional phrase-based hidden Markov model (HMM). However our model presents significant improvements, both in theory and practice.

The model is detailed in section 2, while its EM-based training algorithms are analysed in section 3. Experiments are reported in section 4. Finally, concluding remarks are gathered in section 5.

2 The model

In this section, we present our *phrase-based hidden semi-Markov model (PBHSMM)* for machine translation. Hidden semi-Markov models (HSMMs) (Ostendorf et al., 1996) are a variation on HMM that allow the emission of segments $\boldsymbol{x}_j^{j+l-1}$ at each state instead of constraining the emission to one element x_j as HMM do. Therefore, the probability of emitting an object sequence $\boldsymbol{x}_j^{j+l-1}$ in any state depends on the segment length l. Note that in hidden Markov models (HMMs), the probability of emitting a segment of length l staying in the same state q, can only be simulated by transitions to the same state q. This yields the exponential decaying length probability expressed as follows

$$\mathrm{p}(l \mid q) = [\mathrm{p}(q \mid q)]^{l-1} \quad , \tag{2}$$

which is not appropriate for many situations.

The HSMM model introduced in this section is clearly inspired in the phrase-based translation models (Koehn et al., 2003). The idea behind this model is to provided a well-defined monotonic formalism that, while remaining close to the phrase-based models, explicitly introduces the statistical dependencies needed to define a phrased monotonic translation process. Although the monotonic constraint is an obvious disadvantage for this primer HSMM translation model, it can be extended to non-monotonic processes. However, these extensions lay far beyond the aim of this work.

Albeit there are several ways to formalise a HSMM, we advocate for a similar formalisation of that found in (Murhpy, 2007). Let $\boldsymbol{x} \in \mathcal{X}^\star$ be the source sentence and $\boldsymbol{y} \in \mathcal{Y}^\star$ the target sentence, then we start by decomposing the conditional translation probability, $p(\boldsymbol{x} \mid \boldsymbol{y}, I, J)$. We assume that the monotonic translation process has been carried out from left to right in sequences of words or *phrases*. For this purpose, both sentences should be segmented into the same amount of phrases. Figure 1, depicts an example of a possible monotonic bilingual segmentation in which the source sentence has a length of 9 words, while the target sentence is made up of 11 words. Note that each bilingual phrase makes up a *concept*; for instance c_1, c_2, c_3 and c_4 are concepts in Figure 1. To represent the segmentation process, we use two segmentation variables for both source, $\boldsymbol{l}$, and target, $\boldsymbol{m}$, sentences.

The target segmentation variable $\boldsymbol{m}$ stores each target segment length at the position at which the segment begins. Therefore, if the target segment length variable $\boldsymbol{m}$ has a value greater than 0 at position i, then a segment with length m_i starts at this position i. For instance, the target segmentation represented in Figure 1 is given by $\boldsymbol{m} = \boldsymbol{m}_1^{11} = (3, 0, 0, 3, 0, 0, 2, 0, 3, 0, 0)$. Note that values for the segment length variable such as, $\boldsymbol{m} = (3, 0, 0, 3, 0, 0, 2, 0, 1, 0, 0)$ or $\boldsymbol{m} = (3, 0, 0, 3, 0, 0, 1, 0, 3, 0, 0)$, *are invalid*. It is also worth noting that the domain of the segmentation ranges among all the possible segmentation lengths.

The source segmentation variable $\boldsymbol{l}$ represents the length of each *source segment* at the position at which its corresponding *target segment* ends. If the source segment length variable $\boldsymbol{l}$ has a value greater than 0 at position i; then the length of the source segment corresponding to the target segment that starts at position i, is l_i. For instance, in Figure 1 the source segment length variable is $\boldsymbol{l} = \boldsymbol{l}_1^{11} = (3, 0, 0, 2, 0, 0, 3, 0, 1, 0, 0)$.

Given a target segmentation variable, say $\boldsymbol{m}$, we define its prefix counterpart, $\bar{\boldsymbol{m}}$ as follows

$$\bar{m}_i = \sum_{k=1}^{i} m_k \quad i = 0, 1, \ldots, I \quad . \tag{3}$$

In Figure 1, the prefix segments lengths are $\bar{\boldsymbol{m}}_0^{11} = (0, 3, 3, 3, 6, 6, 6, 8, 8, 11, 11, 11)$ and

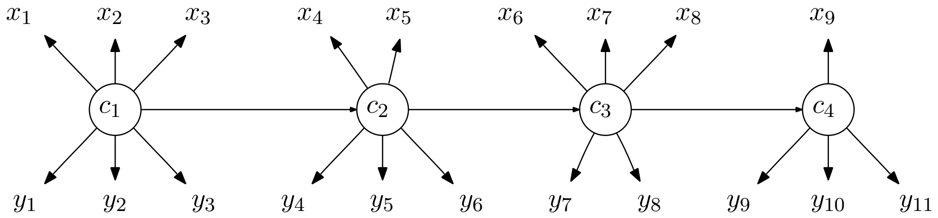

Figure 1: A generative example of the phrase-based hidden semi-Markov model for machine translation.

$\bar{\boldsymbol{l}}_0^{11} = (0, 3, 3, 3, 5, 5, 5, 8, 8, 9, 9, 9)$, for target and source segment length variables respectively.

Mathematically, we express the idea depicted in Figure 1 unhiding the former segmentation length variables

$$p(\boldsymbol{x} \,|\, \boldsymbol{y}) = \sum_{\boldsymbol{l}} \sum_{\boldsymbol{m}} p(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{m} \,|\, \boldsymbol{y}, I, J) \quad . \qquad (4)$$

The completed model in Eq. (4) is decomposed as follows

$$p(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{m} \,|\, \boldsymbol{y}) := p(\boldsymbol{m}) p(\boldsymbol{l} \,|\, \boldsymbol{m}) p(\boldsymbol{x} \,|\, \boldsymbol{m}, \boldsymbol{l}, \boldsymbol{y}) \quad (5)$$

where we have dropped the dependence on $\boldsymbol{y}$ for the segment variables. Note that for clarity we have omitted the dependency on the lengths J and I in all probabilities; and we will henceforth proceed this way.

Both length probabilities in Eq. (5) are being decomposed left-to-right. However, in order to keep the training as fast as possible, a special decomposition of such probabilities is going to be made. We detail here the decomposition of the target segment length probability model, omitting details for the remaining random variables.

The probability of the target segment length variable is given by

$$p(\boldsymbol{m}) = \prod_{i=1}^{I} p(m_i \,|\, \boldsymbol{m}_1^{i-1}) \quad . \qquad (6)$$

At first stage, we had assumed that each partial probability in Eq. (6) does not depend neither on $\boldsymbol{y}$, nor on both lengths (I and J). Hence, the probability $p(m_i \,|\, \boldsymbol{m}_1^{i-1})$ is modelled as follows

$$p(m_i \,|\, \boldsymbol{m}_1^{i-1}) = \begin{cases} \mathrm{p}(m_i) & \bar{m}_{i-1} + 1 = i, \, m_i \neq 0 \\ 1 & \bar{m}_{i-1} + 1 \neq i, \, m_i = 0 \end{cases} \qquad (7)$$

Finally the segment length probability is expressed as follows

$$p(\boldsymbol{m}) := \prod_{i \in \mathcal{Z}(\boldsymbol{m})} 1 \prod_{i \notin \mathcal{Z}(\boldsymbol{m})} \mathrm{p}(m_i) \quad , \qquad (8)$$

where $\mathcal{Z}(\boldsymbol{m})$ or simply $\mathcal{Z}$ stands for the set of positions t for which m_t is 0. For instance, in the example in Figure 1, $\mathcal{Z}$ is instanced to $\mathcal{Z}(\boldsymbol{m}) = \{2, 3, 5, 6, 8, 10, 11\}$.

Provided that one of the two products in Eq. (8) simplifies to 1, the segment length probability is expressed as

$$p(\boldsymbol{m}) := \prod_{i \notin \mathcal{Z}} \mathrm{p}(m_i) \quad . \qquad (9)$$

Since explicitly showing these details forces the discourse to be awkward, we will omit these details. Therefore, we will use equations resembling the following

$$p(\boldsymbol{m}) := \prod_{t} \mathrm{p}(m_t) \quad , \qquad (10)$$

where we have explicitly ommitted that $t \in \mathcal{Z}$, and we have changed the index i into t for subtly summarising the whole previous simplification process. This approach resembles the state probability decomposition in HSMM (Ostendorf et al., 1996).

Similarly to the target segment length model, the source segment length yields the following decomposition

$$p(\boldsymbol{l} \,|\, \boldsymbol{m}) := \prod_{t} \mathrm{p}(l_t \,|\, m_t) \quad . \qquad (11)$$

Finally, knowing the length segment variables, the emission probability is also decomposed left-to-right as follows

$$p(\boldsymbol{x} \,|\, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{y}) := \prod_{t} \mathrm{p}(\boldsymbol{x}(t) \,|\, \boldsymbol{y}(t)) \quad , \qquad (12)$$

where $\boldsymbol{y}(t)$ stands for $\boldsymbol{y}_t^{t+m_t-1}$ and $\boldsymbol{x}(t)$ stands for $\boldsymbol{x}_{\bar{l}_{t-1}+1}^{\bar{l}_t}$; i.e., the t-th "emitted" phrase and its respective t-th target phrase. Note that since t is a boundary of a target segment, then $\bar{l}_t$ is equal to $\bar{l}_{t-1} + l_t$.

Summarising, the proposed (completed) conditional translation model is defined by

$$p(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{m} \mid \boldsymbol{y}) := \prod_{t} \mathrm{p}(m_t)\, \mathrm{p}(l_t \mid m_t)\, \mathrm{p}(\boldsymbol{x}(t) \mid \boldsymbol{y}(t)) \tag{13}$$

Then, the incomplete model introduced in Eq. (4) is parameterised as follows

$$p(\boldsymbol{x}|\boldsymbol{y}) := \sum_{\boldsymbol{l}} \sum_{\boldsymbol{m}} \prod_{t} \mathrm{p}(m_t)\, \mathrm{p}(l_t \mid m_t)\, \mathrm{p}(\boldsymbol{x}(t)|\boldsymbol{y}(t)) \tag{14}$$

with the following parameter set $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = \{\mathrm{p}(m), \mathrm{p}(l \mid m), \mathrm{p}(\boldsymbol{u} \mid \boldsymbol{v})\} \tag{15}$$

where l and m are positive integers, $\boldsymbol{u}$ is a source phrase, i.e., $\boldsymbol{u} \in \mathcal{X}^\star$; and $\boldsymbol{v}$ is a target phrase $\boldsymbol{v} \in \mathcal{Y}^\star$.

It is important to smooth the phrase translation probabilities to avoid over-training. For doing so, we have used the IBM model 1 (Brown and others, 1993) as follows

$$\tilde{\mathrm{p}}(\boldsymbol{u} \mid \boldsymbol{v}) = (1-\epsilon)\ \mathrm{p}(\boldsymbol{u} \mid \boldsymbol{v}) + \epsilon\, p_{IBM1}(\boldsymbol{u} \mid \boldsymbol{v}) \tag{16}$$

Note that in this model, each target phrase $\boldsymbol{y}(t)$ is understood as the "state" of a HSMM in which the source phrase $\boldsymbol{x}(t)$ is emitted. Obviously this is not a pure HSMM in which we have a latent state variable. The omission of this latent variable is more an assumption than a requirement. Recall that in Figure 1 we have depicted each bilingual phrase pair being emitted by a *concept*. Therefore, we could theoretically model this latent variable as well. This inclusion would not significantly change the algorithms proposed here. However, this idea is left as future work, since it is firstly needed to check whether this primer model degrades or not the system performance as some similar works have previously reported (DeNero et al., 2006; Marcu and Wong, 2002).

3 The training

Since the proposed PBHSMM assumes that the segment length variables are not given in the training data, some approximate inference algorithm such as the EM (Dempster et al., 1977) is needed. We omit here the EM derivations which lead to the well-known Baum-Welch algorithm (Rabiner, 1990). This algorithm follows the iterative scheme of all the EM instantiations. First, we guess an adequate parameter set, $\boldsymbol{\theta}^{(0)}$, as a start point. Then, we compute the forward, $\alpha_{tl}^{(0)}(\boldsymbol{x}, \boldsymbol{y})$, and backward, $\beta_{tl}^{(0)}(\boldsymbol{x}, \boldsymbol{y})$, recurrences for each sample. These recurrences are used to compute the fractional counts $\gamma_{tlt'l'}^{(0)}(\boldsymbol{x}, \boldsymbol{y})$; and afterwards, a new $\boldsymbol{\theta}^{(1)}$ is estimated from those fractional counts. The re-estimated parameter set $\boldsymbol{\theta}^{(1)}$ can be used again to re-compute the recurrences, defining an iterative process that ensures the log-likelihood to increase in each iteration (or remain the same). This process goes on until either convergence or a maximum number of iterations is achieved.

3.1 Forward recurrence

The forward recurrence α_{tl} is defined as the prefix probability

$$\alpha_{tl} = \alpha_{tl}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_1^l, \bar{l}_t = l, \bar{m}_t = t \mid \boldsymbol{y}) \tag{17}$$

where $\bar{l}_t = l$ and $\bar{m}_t = t$ mean that a source and a target phrase end/start at position l of the input and t of the output. This prefix probability is recursively computed as follows

$$\alpha_{tl} = \begin{cases} 1 & t = 0, l = 0 \\ \sum_{t'} \sum_{l'} \alpha_{t'l'}\, \mathrm{p}(l' - l, t' - t) & 0 < t \leq I, \\ \qquad \cdot\, \mathrm{p}(\boldsymbol{x}_{l'+1}^{l} | \boldsymbol{y}_{t'+1}^{t}) & 0 < l \leq J \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where the sum over t' ranges from 0 to $t-1$ and likewise the sum over l' ranges from 0 to $l-1$; and where we have used $\mathrm{p}(l'-l, t'-t)$ to denote the product of lengths

$$\mathrm{p}(l' - l, t' - t) = \mathrm{p}(t' - t)\, \mathrm{p}(l' - l \mid t' - t) \quad , \tag{19}$$

in order to compress notation.

3.2 Backward recurrence

The backward recurrence β_{tl} is defined as the following suffix probability

$$\beta_{tl} = \beta_{tl}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{l+1}^{J} | \bar{l}_t = l, \bar{m}_t = t, \boldsymbol{y}) \tag{20}$$

where $\bar{l}_t = l$ and $\bar{m}_t = t$ mean that a source and a target phrase ended/started at position l of the input and t of the output. The former suffix probability is recursively computed as follows

$$\beta_{tl} = \begin{cases} 1 & t = I, l = J \\ \sum_{t'} \sum_{l'} \beta_{t'l'}\, \mathrm{p}(l' - l, t' - t) & 0 \leq t < I, \\ \qquad \cdot\, \mathrm{p}(\boldsymbol{x}_{l+1}^{l'} \mid \boldsymbol{y}_{t+1}^{t'}) & 0 \leq l < J \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where the sum over t' ranges from $t+1$ to I and likewise the sum over l' ranges from $l+1$ to J.

These two recurrences answer the question of which is the probability for a given pair of sentences

$$\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{y}) = \alpha_{IJ} = \beta_{00} \quad . \tag{22}$$

Both the forward and backward recurrence require a matrix of size $O(IJ)$. In order to compute these recurrences a time complexity of $O(I^2J^2)$ is required. However, it can be reduced to $O(IJM^2)$ by defining a maximum phrase length M.

3.3 Fractional counts

Using the previously defined recursions, we can compute the probability of segmenting a given sample in the source positions (l, l') and in the target positions (t, t')

$$\gamma_{tlt'l'} = \frac{\alpha_{tl}\,\mathrm{p}(l'-l, t'-t)\,\mathrm{p}(\boldsymbol{x}_{l+1}^{l'} \mid \boldsymbol{y}_{t+1}^{t'})\beta_{t'l'}}{\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})} \tag{23}$$

This fractional count is very helpful through the Baum-Welch training.

3.4 Re-estimation

Once we have computed the recurrences and the fractional counts, the phrase translation probabilities are re-estimated as follows

$$\mathrm{p}(\boldsymbol{u} \mid \boldsymbol{v}) = \frac{N(\boldsymbol{u}, \boldsymbol{v})}{\sum_{\boldsymbol{u}'} N(\boldsymbol{u}', \boldsymbol{v})} \tag{24}$$

with

$$N(\boldsymbol{u}, \boldsymbol{v}) = \sum_{n} \sum_{l<l'} \sum_{t<t'} \gamma_{ntlt'l'} \delta(\boldsymbol{x}_{l+1}^{l'}, \boldsymbol{u}) \delta(\boldsymbol{y}_{t+1}^{t'}, \boldsymbol{v}) \tag{25}$$

where $\delta(a, b)$ is the Kronecker delta function which is 1 if $a = b$ and 0 otherwise.

The target phrase length probabilities are estimated as follows

$$\mathrm{p}(m) = \frac{N(m)}{\sum_{m'} N(m')} \tag{26}$$

with

$$N(m) = \sum_{n} \sum_{l<l'} \sum_{t} \gamma_{n,t,l,(t+m),l'} \tag{27}$$

Finally, the source phrase length probabilities are re-estimated by

$$\mathrm{p}(l \mid m) = \frac{N(l, m)}{\sum_{l'} N(l', m)} \tag{28}$$

with

$$N(l, m) = \sum_{n} \sum_{l'} \sum_{t} \gamma_{n,t,l,(t+m),(l'+l)} \tag{29}$$

where l denotes a source phrase length, and m a target phrase length.

An alternative training algorithm is obtained computing the maximum segmentation instead of the recurrences. This training, the so-called Viterbi training (Rabiner, 1990), is an iterative training process as well. Each iteration comprises two stages: computing the maximum segmentation and re-estimating the parameters. The Viterbi recursion is used to obtain the maximum segmentation

$$\delta_{tl} = \begin{cases} 1 & t = 0, l = 0 \\ \max_{t',l'} \{\delta_{t'l'}\,\mathrm{p}(l'-l, t'-t) & 0 < t \leq I, \\ \qquad \mathrm{p}(\boldsymbol{x}_{l'+1}^{l} | \boldsymbol{y}_{t'+1}^{t})\} & 0 < l \leq J \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

A traceback of the decisions made to compute δ_{IJ} provides the maximum segmentation $\hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{l}}$.

Afterwards, the re-estimation equations are the similar to Eqs. (24), (26), and (28), but in this case the counts $N(\boldsymbol{u}, \boldsymbol{u})$, $N(m)$, and $N(l, m)$ are the actual counts since the latent segmentation is assumed to be the maximum segmentation.

4 Experiments

The aim of the experimentation is to see how the proposed method and algorithm improves the quality of a any phrase dictionary given as input. For doing so, we have tested our algorithm in two corpora: the Europarl-10 and the Europarl-20. The former comprises all the sentences from the English-to-Spanish part of Europarl (version 3) (Koehn, 2005) with length equal or less than 10. The latter is made up of all the English-to-Spanish Europarl sentences with length equal or less than 20. For both corpora we have randomly selected 5000 sentences for testing the algorithms. Note that we have constrained the training length of the standard Europarl because of the time requirement for training the proposed PBHSMM. Table 1 gathers some basic statistics of the training partition; and Table 2 is the counterpart for testing.

All the experiments were carried out using a 4-gram language model computed with the standard tool SRILM (Stolcke, 2002), and a modified Kneser-Ney smoothing. To define a translation baseline, we compare our results with

Training	Europarl-10		Europarl-20	
	En	Sp	En	Sp
sentences	$76,996$		$306,897$	
avg. length	7.01	7.0	12.6	12.7
running words	546K	540K	3.86M	3.91M
voc. size	16K	22K	37K	58K

Table 1: Basic statistics of the training sets.

Test	Europarl-10		Europarl-20	
	En	Sp	En	Sp
sentences	$5,000$		$5,000$	
avg. length	7.2	7.0	12.8	13.0
running words	35.8K	35.2K	62.1K	63.0K
ppl (3-gram)	53.4	64.4	77.6	86.8

Table 2: Basic statistics of the test sets.

Iterations	En → Sp		Sp → En	
	TER	BLEU	TER	BLEU
Moses $p(\boldsymbol{x} \mid \boldsymbol{y})$ baseline				
	50.0	32.9	47.2	32.7
Iterations	Baum-Welch			
0	51.4	31.9	48.2	33.2
1	51.4	31.9	47.9	33.1
2	51.5	31.9	47.9	33.1
4	51.2	32.6	48.1	33.1
8	51.4	31.8	48.0	33.0
Iterations	Viterbi			
0	51.4	31.9	48.2	33.2
1	51.4	31.9	47.9	33.1
2	51.1	32.6	48.0	33.2
4	51.2	32.6	48.0	33.0
8	51.4	31.8	48.0	33.0

Table 3: Results obtained with the Europarl-10 corpus with a maximum phrase length of 4.

Moses (Koehn and others, 2007) but constraining the model to only use a phrase-based inverse model.

For evaluating the quality of the translations we have used two error measures: bilingual evaluation understudy (BLEU) (Papineni et al., 2001), and translation edit rate (TER) (Snover and others, 2006).

The proposed training algorithms need an initial guess. To this aim, we have computed the IBM word models alignments with GIZA++ (Och and Ney, 2003), for both translation directions. Then, we have computed the simmetrisation heuristic (Och and Ney, 2004) and extracted all the *consistent* phrases (Och and Ney, 2004). Afterwards, we have computed our initial guess by counting the occurrences of each bilingual phrase and then normalising the counts. Instead of directly using the Moses system to do this work, we have implemented our own version of this process.

Since the training algorithm highly depends on the maximum phrase length, for most of the experimentation we have limited it to 4. In Table 3, the results obtained for both translation directions are summarised for the Europarl-10. Surprisingly, Viterbi training obtains almost the same results that the Baum-Welch training; probably because most of the sentences accumulate all the probability mass in just one possible segmentation. Maybe that is why our algorithm is not able to obtain a large improvement with respect to the initialisation. Note that since the proposed system and Moses use different phrase-tables, the comparison of this two numbers is not fair. Therefore, the Moses baseline is only given as a reference and not as a system to improve. The important question is whether the model produces an improvement with respect to the initialisation, i.e., the result on iteration 0. Note that this corpus is small, and although its complexity allow us to check some PBHSMM properties, we cannot to obtain further conclusions.

On the other hand, Table 4 summarises the results obtained with the Europarl-20. This Table only report results for the Viterbi training since again Baum-Welch training has no advantage with respect to it. Typically, over 4 iterations suffices to avoid over-training, and maximise the system performance. The results show a minor improvement over the initialisation. Although the improvement is small, its magnitude is similar to the improvement obtained when extending the maximum phrase length as shown in Table 5. For instance, it is seen that extending the maximum phrase length from 4 to 5 incurs in the same improvement that performing 4 Viterbi iterations with a maximum phrase length of 4. In most of the cases the Viterbi training improves the translation quality.

Although, in most cases the training does not incur in a significant improvement over the baseline; in practice the quality of the translations is increased by the training. In Table 6, we have selected some translation examples. A detailed analysis of the system translations suggest that most cases belong to the cases A or B.

Case A	Training improves evaluation measures
REF.	I sincerely believe that the aim of the present directive is a step in the right direction .
IT. 0	I am convinced that the aim of this directive is a step in the right direction .
IT. 4	I sincerely believe that the aim of the directive before us is a step in the right direction .
MOSES	I sincerely believe that the aim behind the directive is also a step in the right direction .
Case B	Training improves translation but not evaluation measures
REF.	Mr president , i wish to endorse mr posselt 's comments .
IT. 0	Mr president , i support for to our .
IT. 4	Mr president , i join in good faith to our colleague , mr posselt .
MOSES	mr president , i would like to join in good faith in the words of our colleague , mr rbig .
Case C	Training degrades evaluation measures
REF.	BSE has already cost the uk gbp 1.5 billion in lost exports .
IT. 0	BSE has cost the uk 1.5 million losses exports .
IT. 4	BSE already has cost in the uk alone 1500 million pounds into loss of exports .
MOSES	BSE has already claimed to britain 1500 million pounds into loss of trade .
Case D	Other cases
REF.	Are there any objections to amendment nos 3 and 14 being considered as null and void from now on ?
IT. 0	Are there any objections to give amendments nos 3 and 14 .
IT. 4	Are there any objections to adopt amendments nos 3 and 14 ?
MOSES	Are there any objections to consider amendments nos 3 and 14 ?

Table 6: Some translation examples (Sp → En) before and after training the phrase table 4 iterations with the Viterbi training and maximum phrase length of 4.

Iterations	En → Sp		Sp → En	
	TER	BLEU	TER	BLEU
Moses $p(\boldsymbol{x} \mid \boldsymbol{y})$ baseline				
	57.3	23.5	55.1	24.10
Iterations	Viterbi			
0	57.7	25.0	56.0	26.0
1	57.7	25.1	55.8	26.4
2	57.7	25.1	55.9	26.4
4	57.7	25.2	55.8	26.5
8	57.7	25.2	55.8	26.5

Table 4: Results obtained with the Europarl-20 corpus with a maximum phrase length of 4.

5 Conclusions and Future work

We have presented a phrase-based hidden semi-Markov model for machine translation inspired on both phrase-based models and classical hidden semi-Markov models. The idea behind this model is to provide a well-defined monotonic formalism that explicitly introduces the statistical dependencies needed to define the monotonic translation process with theoretical correctness and without moving away from the phrase-based models.

A detailed practical analysis showed a slight improvement by applying the estimation algorithms

Iterations	En → Sp		Sp → En	
	TER	BLEU	TER	BLEU
Iterations	Maximum phrase length 2			
0	60.5	21.2	57.9	23.5
4	60.5	21.2	58.1	23.5
Iterations	Maximum phrase length 3			
0	58.6	24.1	56.1	25.7
4	58.3	24.1	56.4	25.5
Iterations	Maximum phrase length 4			
0	57.7	25.0	56.0	26.0
4	57.7	25.1	55.8	26.5
Iterations	Maximum phrase length 5			
0	57.7	25.1	55.8	26.6
4	57.4	25.3	55.3	26.9
Iterations	Maximum phrase length 6			
0	57.7	25.4	55.9	26.6
4	57.3	25.6	55.4	26.8

Table 5: Results obtained with the Europarl-20 corpus for several maximum phrase lengths.

with respect to the baseline. Surprisingly, we have observed that both trainings, Viterbi and Baum-Welch, obtain the same practical behaviour. Therefore, we recommend the use of the fastest: the Viterbi training. However, we have not used the proposed PBHSMM as a feature inside a log-linear model as most of the current state-of-the-art systems. We leave this comparison as a future work.

As discussed in section 2, one outstanding and simple extension to the proposed model is to unhide the *concept* variable by having a mixture of phrase-based dictionaries, $\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{y}, c)$. Actually, the requirements of this modification would not significantly affect to the proposed estimation algorithms. We are already extending the model towards this direction.

Finally, the most undesirable property of the proposed model is its monotonicity at phrase level. Although the monotonic constraint is a clear disadvantage for this primer PBHSMM translation model, it can be extended to non-monotonic processes. However, we leave these extensions as future work.

Acknowledgement

Work partially supported by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), by the EC (FEDER), the Spanish MEC under grant TIN2006-15694-CO2-01 and the Valencian “Conselleria d’Empresa, Universitat i Ciència” under grant CTBPRA/2005/004.

References

Alexandra Birch, Chris Callison-Burch, Miles Osborne and Philipp Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings of the NAACL 2006 Workshop on Statistical Machine TranslationConference*.

Andrés-Ferrer, J. and A. Juan-Císcar. 2007. A phrase-based hidden markov model approach to machine translation. In *Proceedings of New Approaches to Machine Translation*, pages 57–62, January.

Brown, P. F. et al. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Chen, S. F. and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL’96*, pages 310–318, Morristown, NJ, USA, June. Association for Computational Linguistics.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, 39(1):1–22.

DeNero, J., D. Gillick, J. Zhang, and D. Klein. 2006. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June. Association for Computational Linguistics.

Koehn, P. et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL’07: Demo and Poster Sessions*, pages 177–180, Morristown, NJ, USA, June. Association for Computational Linguistics.

Koehn, P., F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL’03*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. of the MT Summit X*, pages 79–86, September.

Marcu, Daniel and Qilliam Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, Philadelphia, PA, July.

Murhpy, Kevin P. 2007. Hidden semi-Markov Models (HSMMs). Technical report, University of British Columbia.

Och, F.J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, F.J. and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Ostendorf, M., V. Digalakis, and O. A. Kimball. 1996. From hmms to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4:360–378.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176, Thomas J. Watson Research Center.

Rabiner, Lawrence R. 1990. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296.

Snover, M. et al. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA’06*, pages 223–231, Boston, Massachusetts, USA, August. Association for Machine Translation in the Americas.

Stolcke, A. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of ICSLP’02*, pages 901–904, September.