Log-linear weight optimisation via Bayesian Adaptation in Statistical Machine Translation

Germán Sanchis-Trilles and **Francisco Casacuberta**
Departamento de Sistemas Informáticos y Computación
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{gsanchis,fcn}@dsic.upv.es

Abstract

We present an adaptation technique for statistical machine translation, which applies the well-known Bayesian learning paradigm for adapting the model parameters. Since state-of-the-art statistical machine translation systems model the translation process as a log-linear combination of simpler models, we present the formal derivation of how to apply such paradigm to the weights of the log-linear combination. We show empirical results in which a small amount of adaptation data is able to improve both the non-adapted system and a system which optimises the above-mentioned weights on the adaptation set only, while gaining both in reliability and speed.

1 Introduction

The adaptation problem is a very common issue in statistical machine translation (SMT), where it is frequent to have very large collections of bilingual data belonging to e.g. proceedings from international entities such as the European Parliament or the United Nations. However, if we are currently interested in translating e.g. printer manuals or news data, we will need to find a way in which we can take advantage of such data.

The grounds of modern SMT were established in (Brown et al., 1993), where the machine translation problem was defined as follows: given a sentence $\mathbf{f}$ from a certain source language, an equivalent sentence $\hat{\mathbf{e}}$ in a given target language that maximises the posterior probability is to be found. According to the Bayes decision rule, such statement can be specified as follows:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f}) \tag{1}$$

Recently, a direct modelling of the posterior probability $\Pr(\mathbf{e}|\mathbf{f})$ has been widely adopted, and, to this purpose, different authors (Papineni et al., 1998; Och and Ney, 2002) proposed the use of the so-called log-linear models, where

$$p(\mathbf{e}|\mathbf{f}) = \frac{\exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}'} \exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}')} \tag{2}$$

and the decision rule is given by the expression

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}) \tag{3}$$

where $h_k(\mathbf{f}, \mathbf{e})$ is a score function representing an important feature for the translation of $\mathbf{f}$ into $\mathbf{e}$, as for example the language model of the target language, a reordering model or several translation models. K is the number of models (or features) and λ_k are the weights of the log-linear combination. Typically, the weights $\mathbf{\Lambda} = [\lambda_1, \ldots, \lambda_K]^T$ are optimised with the use of a development set.

The use of log-linear models implied an important break-through in SMT, allowing for a significant increase in the quality of the translations produced. In this work, we present a Bayesian technique for adapting the weights of such log-linear models according to a small set of adaptation data.

In this paper, we will be focusing on adapting the weights vector $\mathbf{\Lambda}$, since appropriate values of such vector for a given domain do not necessarily imply a good combination in other domains. One naïve way in which some sort of adaptation can be performed on $\mathbf{\Lambda}$ is to re-estimate these weights

Coling 2010: Poster Volume, pages 1077–1085,
Beijing, August 2010

from scratch only on the adaptation data. However, such re-estimation may not be a good idea, whenever the amount of adaptation data available is not too big. On the one hand, because small amounts of adaptation data may easily yield overtrained values of $\mathbf{\Lambda}$, which may even lead to a degradation of the translation quality. On the other hand, because in some scenarios it is not feasible to re-estimate them because of the time it would take. Moreover, considering a re-estimation of $\mathbf{\Lambda}$ by using both the out-of-domain data and the adaptation set would not be appropriate either. For small amounts of adaptation data, such data would have no impact on the final value of $\mathbf{\Lambda}$, and the time required would be even higher. One such situation may be the Interactive Machine Translation (IMT) paradigm (Barrachina et al., 2009), in which a human translator may start translating a new document, belonging to a specific domain, and the system is required to produce an appropriate output as soon as possible without any prior re-training.

In this paper, a Bayesian adaptation approach solving both problems is presented. Nevertheless, adapting Λ constitutes just a first step towards the adaptation of all the parameters of the SMT model.

The rest of this paper is structured as follows. In next Section, we perform a brief review of current approaches to adaptation and Bayesian learning in SMT. Section 3 describes the typical framework for phrase-based translation in SMT. In Section 4, we present the way in which we apply Bayesian adaptation (BA) to log-linear models in SMT. In Section 5, we describe the practical approximations applied before implementing the BA technique described. In Section 6, experimental design and results are detailed. Conclusions and future work are explained in Section 7.

2 Related work

Adaptation in SMT is a research field that is receiving an increasing amount of attention. In (Nepveu et al., 2004), adaptation techniques were applied to IMT, following the ideas by (Kuhn and Mori, 1990) and adding cache language models (LM) and TMs to their system. In (Koehn and Schroeder, 2007), different ways to combine available data belonging to two different sources was explored; in (Bertoldi and Federico, 2009) similar experiments were performed, but considering only additional source data. In (Civera and Juan, 2007), alignment model mixtures were explored as a way of performing topic-specific adaptation. Other authors (Zhao et al., 2004; Sanchis-Trilles et al., 2009), have proposed the use of clustering in order to extract sub-domains of a large parallel corpus and build more specific LMs and TMs, which are re-combined in test time.

With respect to BA in SMT, the authors are not aware of any work up to the date that follows such paradigm. Nevertheless, there have been some recent approaches towards dealing with SMT from the Bayesian learning point of view. In (Zhang et al., 2008), Bayesian learning was applied for estimating word-alignments within a synchronous grammar.

3 Phrase-based SMT

One of the most popular instantiations of log-linear models in SMT are phrase-based (PB) models (Zens et al., 2002; Koehn et al., 2003). PB models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of PB translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally reorder the translated target phrases in order to compose the target sentence. For this purpose, phrase-tables are produced, in which a source phrase is listed together with several target phrases and the probability of translating the former into the latter. PB models were employed throughout this work.

Typically, the weights of the log-linear combination in Equation 3 are optimised by means of Minimum Error Rate Training (MERT) (Och, 2003). Such algorithm consists of two basic steps. First, n-best hypotheses are extracted for each one of the sentences of a given development set. Next, the optimum $\mathbf{\Lambda}$ is computed so that the best hypotheses in the n-best list, according to a reference translation and a given metric, are ranked higher within such n-best list. These two steps are repeated until convergence.

This approach has two main problems. On the

one hand, that it heavily relies on having a fair amount of data available as development set. On the other hand, that it *only* relies on the data in the development set. These two problems have as consequence that, if the development set made available to the system is not big enough, MERT will most likely become unstable and fail in obtaining an appropriate weight vector $\mathbf{\Lambda}$.

However, it is quite common to have a great amount of data available in a given domain, but only a small amount from the specific domain we are interested in translating. Precisely this scenario is appropriate for BA: under this paradigm, the weight vector $\mathbf{\Lambda}$ is *biased* towards the optimal one according to the adaptation set, while avoiding over-training towards such set by not forgetting the generality provided by the training set. Furthermore, recomputing $\mathbf{\Lambda}$ from scratch by means of MERT may imply a computational overhead which may not be acceptable in certain environments, such as SMT systems configured for online translation, IMT or Computer Assisted Translation, in which the final human user is waiting for the translations to be produced.

4 Bayesian adaptation for SMT

The main idea behind Bayesian learning (Duda et al., 2001; Bishop, 2006) is that model parameters are viewed as random variables having some kind of a priori distribution. Observing these random variables leads to a posterior density, which typically peaks at the optimal values of these parameters. Following the notation in Equation 1, previous statement is specified as

$$p(\mathbf{e}|\mathbf{f};T) = \int p(\mathbf{e},\theta|\mathbf{f};T)d\theta \quad (4)$$

where T represents the complete training set and θ are the model parameters.

However, since we are interested in Bayesian *adaptation*, we need to consider one training set T and one adaptation set A, leading to

$$p(\mathbf{e}|\mathbf{f};T,A) \approx \int p(\theta|T,A)p(\mathbf{e}|\mathbf{f},\theta)d\theta \quad (5)$$

In Equation 5, the integral over the complete parametric space forces the model to take into account all possible values of the model parameters, although the prior over the parameters implies that our model will prefer parameter values which are closer to our prior knowledge. Two assumptions have been made: first, that the output sentence $\mathbf{e}$ only depends on the model parameters (and not on the complete training and adaptation data). Second, that the model parameters do not depend on the actual input sentence $\mathbf{f}$. Such simplifications lead to a decomposition of the integral in two parts: the first one, $p(\theta|T,A)$ will assess how good the current model parameters are, and the second one, $p(\mathbf{e}|\mathbf{f},\theta)$, will account for the quality of the translation $\mathbf{e}$ given the current model parameters.

Then, the decision rule given in Equation 1 is redefined as

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}|\mathbf{f};T,A) \quad (6)$$

Operating with the probability of θ, we obtain:

$$p(\theta|T,A) = \frac{p(A|\theta;T)\,p(\theta|T)}{\int p(A|\theta)\,p(\theta|T)\,d\theta} \quad (7)$$

$$p(A|\theta;T) = \prod_{\forall a\in A} p(\mathbf{f}_a|\theta)\,p(\mathbf{e}_a|\mathbf{f}_a,\theta) \quad (8)$$

where the probability of the adaptation data has been assumed to be independent of the training data and has been modelled as the probability of each bilingual sample $(\mathbf{f}_a,\mathbf{e}_a) \in A$ being generated by our translation model.

Assuming that the model parameters depend on the training data and follow a normal distribution, we obtain

$$p(\theta|T) = \frac{1}{(2\pi)^{d/2}}\exp\left\{-\frac{1}{2}(\theta-\theta_T)^T(\theta-\theta_T)\right\} \quad (9)$$

where θ_T is the set of parameters estimated on the training set and the variance has been assumed to be bounded for all parameters. d is the dimensionality of θ.

Lastly, assuming that our translation model is a log-linear model as described in Equation 3 and that the only parameters we want to adapt are the log-linear weights:

$$p(\mathbf{e}|\mathbf{f},\theta) = \frac{\exp\sum_k \lambda_k\, f_k(\mathbf{f},\mathbf{e})}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k\, f_k(\mathbf{f},\mathbf{e}')} \quad (10)$$

where the model parameters θ have been instantiated to include only the log-linear weights $\mathbf{\Lambda}$.

Finally, combining Equations 8, 9 and 10, and considering only as model parameters the log-linear weights, we obtain:

$$p(\mathbf{e}|\mathbf{f};T,A) = \mathcal{Z}\int p(A|\mathbf{\Lambda};T)p(\mathbf{\Lambda}|T)p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})d\mathbf{\Lambda}$$

$$= \mathcal{Z}\int \prod_{\forall a\in A} \frac{\exp\sum_k \lambda_k\ f_k(\mathbf{f}_a,\mathbf{e}_a)}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k\ f_k(\mathbf{f}_a,\mathbf{e}')}\cdot$$

$$\exp\left\{-\frac{1}{2}(\mathbf{\Lambda}-\mathbf{\Lambda}_T)^T(\mathbf{\Lambda}-\mathbf{\Lambda}_T)\right\}\cdot$$

$$\frac{\exp\sum_k \lambda_k\ f_k(\mathbf{f},\mathbf{e})}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k\ f_k(\mathbf{f},\mathbf{e}')}\ d\mathbf{\Lambda} \qquad (11)$$

where $\mathcal{Z}$ is the denominator present in the previous equation and may be factored out because it does not depend on the integration variable. It has also been assumed that $p(\mathbf{f}_a|\theta)$ is uniform and can also be factored out.

5 Practical approximations

Although the integral described in Equation 11 is the right thing to do from the theoretical point of view, there are several issues which need to be treated first before implementing it.

Since computing the integral over the complete parametric space is computationally impossible in the case of SMT, we decided to perform a Monte Carlo like sampling of these parameters by assuming that the parameters follow a normal distribution centred in $\mathbf{\Lambda}_T$, the weight vector obtained from the training data. This sampling was done by choosing alternatively only one of the weights in $\mathbf{\Lambda}_T$, modifying it randomly within a given interval, and re-normalising accordingly. Equation 11 is approximated in practise as

$$p(\mathbf{e}|\mathbf{f};T,A) = \sum_{\Lambda_m\in MC(\Lambda_T)} p(A|\mathbf{\Lambda};T)p(\mathbf{\Lambda}|T)p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})$$

where $MC(\Lambda_T)$ is the set of Λ_m weights generated by the above-mentioned procedure.

There is still one issue when trying to implement Equation 11. The denominator within the components $p(A|\mathbf{\Lambda};T)$ and $p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})$ contains a sum over all possible sentences of the target language, which is not computable. For this reason, $\sum_{\mathbf{e}'}$ is approximated as the sum over all the hypothesis within a given n-best list. Moreover, instead of performing a full search of the best possible translation of a given input sentence, we will perform a rerank of the n-best list provided by the decoder according to Equation 11.

Typical state-of-the-art PB SMT systems do not guarantee complete coverage of all possible sentence pairs due to the great number of heuristic decisions involved in the estimation of the translation models. Moreover, out-of-vocabulary words may imply that the SMT model is unable to explain a certain bilingual sentence completely. Hence, $p(A|\mathbf{\Lambda};T)$ is approximated as

$$p(A|\mathbf{\Lambda};T) \approx \prod_{\forall a\in A} \frac{\exp\sum_k \lambda_k\ f_k(\mathbf{f}_a,\mathbf{e}_a^*)}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k f_k(\mathbf{f}_a,\mathbf{e}')} \qquad (12)$$

where $\mathbf{e}^*$ represents the best hypothesis the search algorithm is able to produce, according to a given translation quality measure. As in Equation 11, $p(\mathbf{f}_a|\theta)$ has been assumed uniform.

Once the normalisation factor within Equation 7 has been removed, and the above-mentioned approximations have been introduced, $p(\mathbf{e}|\mathbf{f};T,A)$ is no longer a probability. This fact cannot be underestimated, since it means that the terms $p(A|\mathbf{\Lambda};T)$ and $p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})$ on the one hand, and $p(\mathbf{\Lambda}|T)$ on the other, may have very different numeric ranges. For this reason, and in order to weaken the influence of this fact, we introduce a leveraging term δ, such that

$$p(\mathbf{e}|\mathbf{f};T,A) = \sum_{\Lambda_m\in MC(\Lambda_T)} \left(p(A|\mathbf{\Lambda};T)p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})\right)^{\frac{1}{\delta}} p(\mathbf{\Lambda}|T) \qquad (13)$$

Although there are other, more standard, ways of adding this leveraging term, we chose this one for numeric reasons.

6 Experiments

6.1 Experimental setup

Translation quality will be assessed by means of BLEU and TER scores. BLEU measures n-gram precision with a penalty for sentences that are too short (Papineni et al., 2001), whereas TER (Snover et al., 2006) is an error metric that

		Spanish	English
Training	Sentences	731K	
	Run. words	15.7M	15.2M
	Vocabulary	103K	64K
Development	Sentences	2K	
	Run. words	61K	59K
	OoV words	208	127

Table 1: Main figures of the Europarl corpus. *OoV* stands for Out of Vocabulary. K/M stands for thousands/millions of elements.

		Spanish	English
Test 2008	Sentences	2051	
	Run. words	50K	53K
	OoV. words	1247	1201
Test 2010	Sentences	2489	
	Run. words	62K	66K
	OoV. words	1698	1607

Table 2: Main figures of the News-Commentary test sets. *OoV* stands for Out of Vocabulary words with respect to the Europarl corpus.

computes the minimum number of edits required to modify the system hypotheses so that they match the references. Possible edits include insertion, deletion, substitution of single words and shifts of word sequences.

For computing $\mathbf{e}^*$ as described in Equation 12, TER was used, since BLEU implements a geometrical average which is zero whenever there is no common 4-gram between reference and hypothesis. Hence, it is not well suited for our purposes since the complete set of n-best candidates provided by the decoder can score zero.

As a first baseline system, we trained a SMT system on the Europarl Spanish–English training data, in the partition established in the Workshop on SMT of the NAACL 2006 (Koehn and Monz, 2006), using the training and development data provided that year. The Europarl corpus (Koehn, 2005) is built from the transcription of European Parliament speeches published on the web. Statistics are provided in Table 1.

We used the open-source MT toolkit Moses (Koehn et al., 2007)[1] in its default monotonic setup, and estimated the weights of the log-linear combination using MERT on the Europarl development set. A 5-gram LM with interpolation and Kneser-Ney smoothing (Kneser and Ney, 1995) was also estimated.

Since our purpose is to adapt the initial weight vector obtained during the training stage (i.e. the one obtained after running MERT on the Europarl development set), the tests sets provided for the 2008 and 2010 evaluation campaigns of the above-mentioned workshop (Table 2) were also used. These test sets, unlike the one provided in 2006, were extracted from a news data corpus, and can be considered out of domain if the system has been trained on Europarl data.

All the experiments displaying BA results were carried out by sampling a total of 100 random weights, according to preliminary investigation, following the procedure described in Section 5. For doing this, one single weight was added a random amount between 0.5 and -0.5, and then the whole $\mathbf{\Lambda}$ was re-normalised.

With the purpose of providing robustness to the results, every point in each plot of this paper constitutes the average of 10 repetitions, in which the adaptation data was randomly drawn from the News-Commentary test set 2008.

6.2 Comparison between BA and MERT

The effect of increasing the number of adaptation samples made available to the system was investigated. The adaptation data was used either for estimating $\mathbf{\Lambda}$ using MERT, or as adaptation sample for our BA technique. Results can be seen in Figure 1. The δ scaling factor described in Equation 13 was set to 8. As it can be seen, the BA adaptation technique is able to improve consistently the translation quality obtained by the non-adapted system, both in terms of BLEU and TER. These improvements are quite stable even with as few as 10 adaptation samples. This result is very interesting, since re-estimating $\mathbf{\Lambda}$ by means of MERT is only able to yield improvements when provided with at least 100 adaptation samples, displaying a very chaotic behaviour until that point.

In order to get a bit more insight about this chaotic behaviour, confidence interval sizes are shown in Figure 2, at a 95% confidence level, resulting of the repetitions described above. MERT yields very large confidence intervals (as large as 10 TER/BLEU points for less than 100 samples), turning a bit more stable from that point on, where the size of the confidence interval converges slowly to 1 TER/BLEU point. In contrast,

[1] Available from http://www.statmt.org/moses/

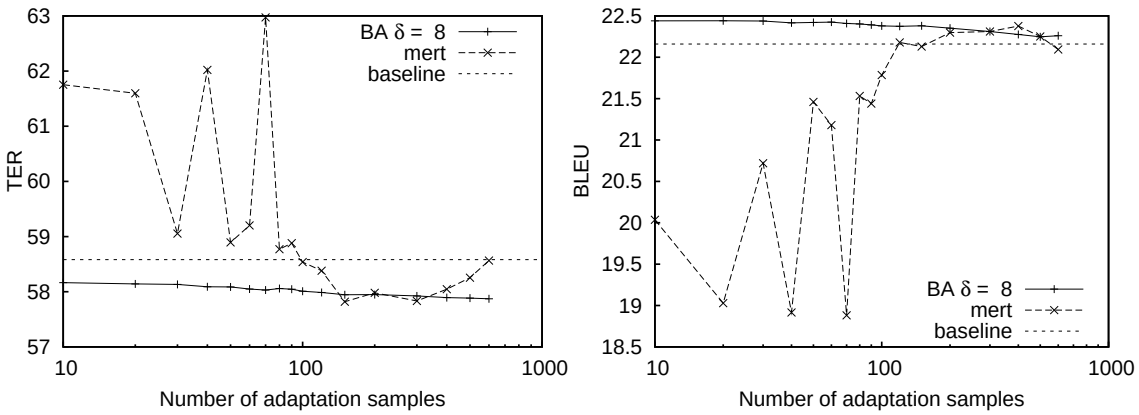

Figure 1: Comparison of translation quality, as measured by BLEU and TER, for baseline system, adapted systems by means of BA and MERT. Increasing number of samples is considered.

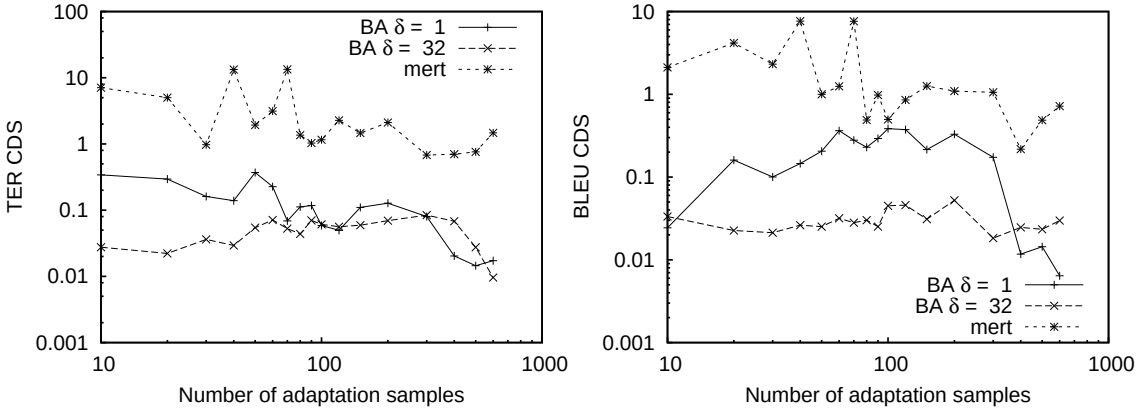

Figure 2: Confidence interval sizes (CDS) for MERT and two BA systems, for different number of adaptation samples. For visibility purposes, both axes are in logarithmic scale.

our BA technique yields very small confidence intervals, about half a TER/BLEU point in the worst case, with only 10 adaptation samples. This is worth emphasising, since estimating $\mathbf{\Lambda}$ by means of MERT when very few adaptation data is available may improve the final translation quality, but may also degrade it to a much larger extent. In contrast, our BA technique shows stable and reliable improvements from the very beginning. Precisely under such circumstances is an adaptation technique useful: when the amount of adaptation data is small. In other cases, the best thing one can do is to re-estimate the model parameters from scratch.

Example translations, extracted from the experiments detailed above, are shown in Figure 5.

6.3 Varying δ

So as to understand the role of scaling factor δ, results obtained varying it are shown in Figure 3. Several things should be noted about these plots:

- Increasing δ leads to smoother adaptation curves. This is coherent with the confidence interval sizes shown in Figure 1.
- Smaller values of δ lead to a slight degradation in translation quality when the amount of adaptation samples becomes larger. The reason for this can be explained by looking at Equation 13. Since $p(A|\mathbf{\Lambda};T)$ is implemented as a product of probabilities, the more adaptation samples the smaller becomes $p(A|\mathbf{\Lambda};T)$, and a higher value of δ is needed to compensate this fact. This suggests the need of a δ which depends on the size of the adaptation sample.
- Larger values of δ do not suffer the problem described above, but yield smaller improvements in terms of translation quality for smaller amount of samples.

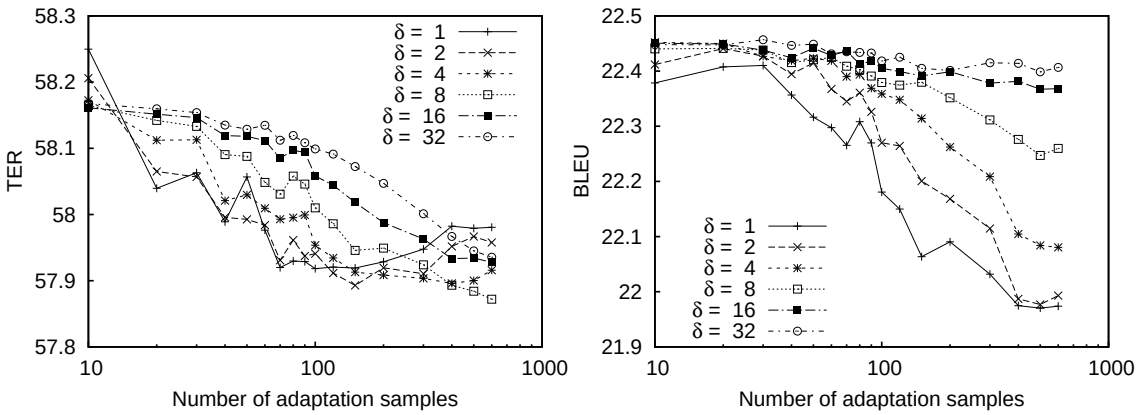

Figure 3: Translation quality comparison for different δ values and number of adaptation samples.

It might seem odd that translation quality as measured by BLEU drops almost constantly as the number of adaptation samples increases. However, it must be noted that the BA technique implemented is set to optimise TER, and not BLEU. Analysing the BLEU scores obtained, we realised that the n-gram precision does increase, but the final BLEU score drops because of a worsening brevity penalty, which is not taken into account when optimising the TER score.

6.3.1 Increasing the n-best order

The effect of increasing the order of n-best considered was also analysed. In order to avoid an overwhelming amount of results, only those obtained when considering 100 adaptation samples are displayed in Figure 4. As it can be seen, TER drops monotonically for all δ values, until about 800, where it starts to stabilise. Similar behaviour is observed in the case of BLEU, although depending on δ the curve shows an improvement or a degradation. Again, this is due to the brevity penalty, which TER does not implement, and which induces this inverse correlation between TER and BLEU when optimising TER.

7 Conclusions and future work

We have presented a Bayesian theoretical framework for adapting the parameters of a SMT system. We have derived the equations needed to implement BA of the log-linear weights of a SMT system, and present promising results with a state-of-the-art SMT system using standard corpora in SMT. Such results prove that the BA framework can be very effective when adapting the mentioned weights. Consistent improvements are obtained over the baseline system with as few as 10 adaptation samples. The BA technique implemented is able to yield results comparable with a complete re-estimation of the parameters even when the amount of adaptation data is sufficient for such re-estimation to be feasible. Experimental results show that our adaptation technique proves to be much more stable than MERT, which relies very heavily on the amount of adaptation data and turns very unstable whenever few adaptation samples are available. It should be emphasised that an adaptation technique, by nature, is only useful whenever few adaptation data is available, and our technique proves to behave well in such context.

Intuitively, the BA technique presented needs first to compute a set of random weights, which are the result of sampling a gaussian distribution whose mean is the best weight vector obtained in training. Then, each hypothesis of a certain test source sentence is rescored according to the following three components:

- The probability of the adaptation corpus under each specific random weight
- The probability of such random weight according to a prior over the weight vector
- The probability of the current hypothesis under those weights

Concerning computational time, our adaptation technique can easily be implemented within the decoder itself, without any significant increase in computational complexity. We consider this im-

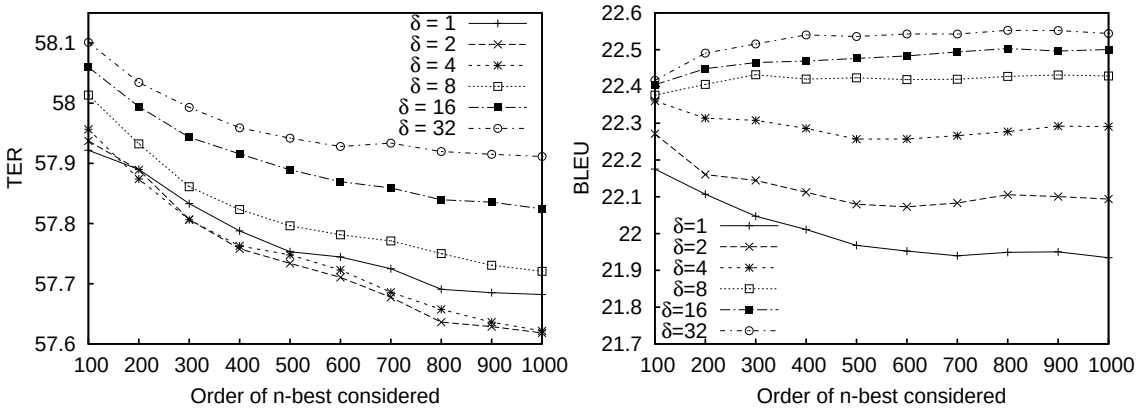

Figure 4: Translation quality for different δ values and n-best sizes considered in the BA system.

source	en afganistán , barack obama espera que se repita el milagro .
reference	barack obama hopes that , in afghanistan , the miracle will repeat itself .
baseline	in afghanistan , barack obama waiting to be repeated the miracle .
BA s10	in afghanistan , barack obama expected to repeat the miracle .
BA s600	in afghanistan , barack obama expected to repeat the miracle .
MERT s10	in afghanistan , barack obama expected to repeat of the miracle .
MERT s600	in afghanistan , barack obama hopes that a repetition of the miracle .
source	al final todo fue más rpido de lo que se pensó .
reference	it all happened a lot faster than expected .
baseline	at the end of all was more quickly than we thought .
BA s10	ultimately everything was more quickly than we thought .
BA s600	ultimately everything was more quickly than we though .
MERT s10	the end all was quicker than i thought .
MERT s600	ultimately everything was quicker than i thought .

Figure 5: Example of translations found in the corpus. `s10` means that only 10 adaptation samples were considered, whereas `s600` means that 600 were considered.

portant, since it implies that rerunning MERT for each adaptation set is not needed, and this is important whenever the final system is set up in an on-line environment.

The derivation presented here can be easily extended in order to adapt the feature functions of the log-linear model (i.e. not the weights). This is bound to have a more important impact on translation quality, since the amount of parameters to be adapted is much higher. We plan to address this issue in future work.

In addition, very preliminary experiments show that, when considering reordering, the advantages described here are larger.

A preliminary version of the present paper was accepted at the Joint IAPR International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition 2010. The main contributions of the present paper constitute more extensive experiments, which have been conducted on standard SMT corpora. Furthermore, in this paper we present the results of adding the leveraging term δ, of applying a random, Monte-Carlo like weight sampling (which was not done previously), and an extensive analysis of the effect of varying the order of n-best considered.

We also plan to implement Markov Chain Monte Carlo for sampling the parameters, and analyse the effect of combining the in-domain and out of domain data for MERT. Such results were not included here for time constraints.

Acknowledgments

This paper is based upon work supported by the EC (FEDER/FSE) and the Spanish MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018) and the iTrans2 (TIN2009-14511) project. Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Generalitat Valenciana under grant Prometeo/2009/014.

The authors would like to thank the anonimous reviewers for their constructive comments.

References

Barrachina, S., O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda H. Ney, J. Tomás, and E. Vidal. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Bertoldi, N. and M. Federico. 2009. Domain adaptation in statistical machine translation with monolingual resources. In *Proc. of EACL WMT*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Brown, P.F., S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of machine translation. In *Computational Linguistics*, volume 19, pages 263–311, June.

Civera, J. and A. Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *Proc. of ACL WMT*.

Duda, R., P. Hart, and D. Stork. 2001. *Pattern Classification*. Wiley-Interscience.

Kneser, R. and H. Ney. 1995. Improved backing-off for m-gram language modeling. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, II:181–184, May.

Koehn, P. and C. Monz, editors. 2006. *Proc. on the Workshop on SMT*. Association for Computational Linguistics, June.

Koehn, P. and J. Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proc. of ACL WMT*.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT/NAACL'03*, pages 48–54.

Koehn et al., P. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Kuhn, R. and R. De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on PAMI*, 12(6):570–583.

Nepveu, L., G. Lapalme, P. Langlais, and G. Foster. 2004. Adaptive language and translation models for interactive machine translation. In *Proc. of EMNLP*.

Och, F. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the ACL'02*, pages 295–302.

Och, F.J. 2003. Minimum error rate training for statistical machine translation. In *Proc. of Annual Meeting of the ACL*, July.

Papineni, K., S. Roukos, and T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. of ICASSP*, pages 189–192.

Papineni, K., A. Kishore, S. Roukos, T. Ward, and W. Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022)*.

Sanchis-Trilles, G., M. Cettolo, N. Bertoldi, and M. Federico. 2009. Online Language Model Adaptation for Spoken Dialog Translation. In *Proc. of IWSLT*, Tokyo.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA'06*.

Zens, R., F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Proc. of KI'02*, pages 18–32.

Zhang, Hao, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.

Zhao, B., M. Eck, and S. Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proc. of CoLing*.