Bayesian Adaptation for Statistical Machine Translation

Germán Sanchis-Trilles and Francisco Casacuberta

Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
{gsanchis,fcn}@dsic.upv.es

Abstract. In many pattern recognition problems, learning from training samples is a process that requires important amounts of training data and a high computational effort. Sometimes, only limited training data and/or limited computational resources are available, but there is also available a previous system trained for a closely related task and with enough training material. This scenario is very frequent in statistical machine translation and adaptation can be a solution to deal with this problem. In this paper, we present an adaptation technique for (state-of-the-art) log-linear modelling based on the well-known Bayesian learning paradigm. This technique has been applied to statistical machine translation and can be easily extended to other pattern recognition areas in which log-linear models are used. We show empirical results in which a small amount of adaptation data is able to improve both the non-adapted system and a system that optimises the above-mentioned weights only on the adaptation set.

1 Introduction

Adaptation in pattern recognition is the task of porting a system trained on a specific task or domain so that it can be used in a different environment. This problem is particularly challenging in natural language processing and other fields where the process of acquiring labelled training samples from a specific domain or task is very costly, but a large collection of labelled data from a similar task is already available. Hence, the challenge consists in being able to modify the original models in such a way, that we are able to take advantage of such large amounts of data available while having at our disposal only very limited amounts of adaptation data.

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

The grounds of modern SMT, a pattern recognition approach to machine translation, were established in [1], where the problem of machine translation

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 620–629, 2010.

was defined as follows: given a sentence $\boldsymbol{x}$ from a certain source language, an equivalent sentence $\hat{\boldsymbol{y}}$ in a given target language that maximises the posterior probability is to be found. Such a statement can be specified, according to the Bayes decision rule, as follows:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \tag{1}$$

Recently, a direct modelling of the posterior probability $Pr(\boldsymbol{y}|\boldsymbol{x})$ has been widely adopted, and, to this purpose, different authors [2,3] proposed the use of the so-called log-linear models, where

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y}')} \tag{2}$$

and the decision rule is given by the expression

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y}) \tag{3}$$

where $h_k(\boldsymbol{x}, \boldsymbol{y})$ is a score function representing an important feature for the translation of $\boldsymbol{x}$ into $\boldsymbol{y}$, 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 $\boldsymbol{\Lambda} = \lambda_1 \ldots \lambda_K$ are optimised with the use of a development set.

Log-linear models implied an important break-through in SMT, allowing for a significant increase in translation quality. In addition, log-linear models have also been applied successfully in other pattern recognition tasks, such as text recognition [4] and speech recognition [5]. 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. Such technique, although applied to SMT in the current paper, is easily extensible to other fields were log-linear models are used.

The rest of this paper is structured as follows. In the next Section, we perform a brief review of current approaches to adaptation and Bayesian learning in SMT. Section 3 describes the typical procedure for weight optimisation 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, experimental design and experimental results are detailed. Finally, conclusions and future work are explained in Section 6.

2 Related Work

Adaptation in SMT is a research field that is receiving an increasing amount of attention. One of the first approaches to this task was performed by [6], in which the translation model (TM) is implemented as an unsupervised multinomial mixture of TMs, where each one was supposed to concentrate most of its probability mass in a certain topic. Later, [7] applied other adaptation techniques to interactive machine translation, following the ideas by [8] and adding cache language

models (LM) and TMs to their system. In [9], different ways to combine available data belonging to two different sources was explored; in [10] similar experiments were performed, but considering only additional source data. In [11], alignment model mixtures were explored as a way of performing topic-specific adaptation, the alignments being used only to extract phrases. Finally, other authors [12,13], have proposed the use of clustering in order to extract the 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, such as [14], in which Bayesian learning is applied in order to estimate appropriate word-alignments within a synchronous grammar.

3 Weight Optimisation in SMT

One of the most popular instantiations of log-linear models in SMT are phrase-based models [15,16]. Phrase-based models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of phrase-based translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally to 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. Phrase-based models were employed throughout this work.

Typically, the weights $\boldsymbol{\Lambda}$ of the log-linear combination in Equation 3 $\boldsymbol{\Lambda}$ are optimised by means of Minimum Error Rate Training (MERT) [17]: first, n-best hypotheses are extracted for each one of the sentences of a given development set. Next, the optimum $\boldsymbol{\Lambda}$ is computed so that the best hypotheses in the n-best list, according to a reference translation and a given metric, are the ones that the search algorithm would produce. These two steps are repeated until convergence, where the weight vector $\boldsymbol{\Lambda}$ remains unchanged.

This approach has two main problems. On the one hand, it heavily relies on having a fair amount of data available as development set. On the other hand, 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 $\boldsymbol{\Lambda}$. In addition, running MERT in systems where the user is waiting actively for the translation to be produced may not be acceptable.

However, it is quite common to have a great amount of data available in a given domain, but only a small amount of data available from the domain we are interested in translating. Precisely this scenario is appropriate for BA: under this paradigm, the weight vector $\boldsymbol{\Lambda}$ is *biased* towards the optimal one according to the adaptation set. However, over-training towards such set is avoided by not forgetting the generality provided by the training set.

4 Bayesian Adaptation for SMT

The main idea behind Bayesian learning is that parameters are viewed as random variables which have some kind of a priori distribution. In such case, observing these random variables leads to a posterior density, which typically peaks at the optimal values of these parameters. Following the notation in 1, the previous statement can be specified as

$$p(\boldsymbol{y}|\boldsymbol{x};T) = \int p(\boldsymbol{y},\theta|\boldsymbol{x};T)d\theta \tag{4}$$

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

Since in this case we are interested in Bayesian *adaptation*, we need to consider one training set T and one adaptation set A, leading to

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x};T,A) &= \int p(\boldsymbol{y},\theta|\boldsymbol{x};T,A)d\theta \\ &= \int p(\theta|T,A)p(\boldsymbol{y}|\boldsymbol{x},\theta)d\theta \end{aligned} \tag{5}$$

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 $\boldsymbol{y}$ only depends on the model parameters (not on the complete training and adaptation data), and second, that model parameters do not depend on the actual input sentence $\boldsymbol{x}$. Such simplifications lead to a decomposition of the integral into two parts: the first one, $p(\theta|T,A)$ will assess how good the current model parameters are, and the second one, $p(\boldsymbol{y}|\boldsymbol{x},\theta)$, will account for the quality of the translation $\boldsymbol{y}$ given the current model parameters.

Operating with the probability of the model parameters, we obtain:

$$p(\theta|T,A) = \frac{p(A|\theta;T)\ p(\theta|T)}{\int p(A|\theta)\ p(\theta|T)\ d\theta} \tag{6}$$

$$p(A|\theta;T) = p(A|\theta) = \prod_{\forall a \in A} p(\boldsymbol{x}_a|\theta)\ p(\boldsymbol{y}_a|\boldsymbol{x}_a,\theta) \tag{7}$$

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 $(\boldsymbol{x}_a, \boldsymbol{y}_a) \in A$ being generated by our translation model.

Assuming that the model parameters follow a normal distribution, we obtain

$$p(\theta|T) = \frac{1}{(2\pi)^{-\sigma_T/2}|\sigma_T|^{-1/2}}\ \exp\left\{-\frac{1}{2}(\theta-\theta_T)^T\sigma_T^{-1}(\theta-\theta_T)\right\} \tag{8}$$

where θ_T is the set of parameters estimated on the training set and σ_T is the variance, which has been assumed to be bounded for all parameters.

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

$$p(\boldsymbol{y}|\boldsymbol{x},\theta) = \frac{\exp \sum_k \theta_k \ f_k(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \ f_k(\boldsymbol{x},\boldsymbol{y}')} \tag{9}$$

Finally, combining Equations 7, 8 and 9, yields:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x};T,A) &= \int \frac{p(A|\theta;T)\ p(\theta|T)}{\int p(A|\theta)\ p(\theta|T)\ d\theta}\ p(\boldsymbol{y}|\boldsymbol{x},\theta)\ d\theta \\ &= \mathcal{Z} \int \prod_{\forall a \in A} p(\boldsymbol{x}_a|\theta)\ p(\boldsymbol{y}_a|\boldsymbol{x}_a,\theta)\ \mathcal{N}(\theta;\theta_T,\sigma_T) p(\boldsymbol{y}|\boldsymbol{x},\theta)\ d\theta \qquad (10) \\ &= \mathcal{Z}' \int \prod_{\forall a \in A} \frac{\exp \sum_k \theta_k \ f_k(\boldsymbol{x}_a,\boldsymbol{y}_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \ f_k(\boldsymbol{x}_a,\boldsymbol{y}')} \qquad (11) \\ &\qquad \exp\left\{-\frac{1}{2}(\theta-\theta_T)^T \sigma_T^{-1}(\theta-\theta_T)\right\} \frac{\exp \sum_k \theta_k \ f_k(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \ f_k(\boldsymbol{x},\boldsymbol{y}')}\ d\theta \end{aligned}$$

where, in Equation 10, $\mathcal{Z}$ is the denominator present in the previous equation and may be out-factored because it does not depend on the integration variable. In Equation 11, it has been assumed that the probability of the input sentence does not depend on the model parameters, and hence it can also be out-factored.

5 Experiments

In this section we will detail the experiments carried out. We will first train a SMT system on training data, and then we will analyse the performance of such system when used for translating data which does not belong to the same domain as the training data. We will follow two adaptation procedures. On the one hand, log-linear model weights are estimated on the adaptation data, forgetting about the estimates obtained in training time. On the other hand, we will perform experiments with our BA technique, and finally compare both approaches.

5.1 Experimental Setup

In this work, we will be assessing translation quality by means of two standard scoring metrics in SMT, namely BLEU and TER scores. BLEU measures the precision of n-grams [18] with a penalty for too short sentences, whereas TER [19] is an error metric that 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.

To train the baseline system, we used the Europarl corpus [20], with the partition established for the Workshop of SMT of NAACL 2006 [21]. Specifically, we performed experiments on Spanish–English translation. The corpus Europarl corpus is divided into three separate sets: one for training, one for development and one for test. The figures of the Europarl corpus are shown in Table 1.

Table 1. Main figures of the Europarl corpus. *OoV* stands for Out of Vocabulary.

		Spanish	English
Training	Sentences	731K	
	Run. words	15.7M	15.2M
	Vocabulary	103K	64K
Development	Sentences	2000	
	Run. words	61K	59K
	OoV words	208	127
Test	Sentences	2000	
	Run. words	60K	58K
	OOV words	207	125

Table 2. Main figures of the Xerox and EU corpora. *OoV* stands for Out of Vocabulary.

		Xerox		EU	
		Spanish	English	Spanish	English
Training	Sentences	55K		164K	
	Run. words	712K	631K	3.4M	3.1M
	Vocabulary	11K	8K	45M	34M
Test	Sentences	1120		800	
	Run. words	10K	8K	23K	20K
	OoV words	42	27	97	81
	OoV w.r.t. Europarl	131	139	156	178
	ppl w.r.t. Europarl	2555	9595	130	194

Since we will be performing adaptation, we also used two other corpora, namely the Xerox corpus [22] and the *EU* corpus [23]. The Xerox corpus is a compendium of user manuals for Xerox printers and photocopiers and was translated from English into other languages by Xerox's language services. The EU corpus was built from the Bulletin of the European Union and is publicly available on the Internet. In this paper, we will focus on the Spanish–English sub-corpora. These two corpora are divided into two separate subsets, one for training and one for test. Their characteristics can be seen in Table 2. It must be noted that EU and Europarl corpora belong to very similar domains, whereas Xerox belongs to a very different domain. This fact is the reason why the Xerox corpus reports such high perplexity (ppl) rates with respect to a language model estimated on the Europarl corpus. Intuitively, perplexity measures how "surprised" the language model is when provided a given test set, i.e. how different such set is with respect to the data it was trained on.

We conducted our experiments by means of the Moses toolkit [24], which implements a statistical log-linear model including five translation scores, a language model, a distortion model, and word and phrase penalties. The five translation scores included provide standard direct and inverted frequency-based and lexical-based probabilities for each phrase pair in the phrase-table.

The initial weights for the log-linear model were estimated by means of MERT on the Europarl development set, as is typically done in SMT. The score to be optimised in this case was BLEU.

5.2 Practical Approximations

In order to find the best scoring sentence according to Equation 11, we asked the decoder to output a list of 500-best for each one of the translated sentences. Such n-best list was then re-ranked according to the score provided by Equation 11 after dropping the normalisation factor Z', since such factor is constant when choosing the maximum scoring output sentence.

Since a true integral over all possible weights is not feasible for computational reasons, we discretised the integral to consider only a set of sample weight vectors. Here, such sampling was performed by taking into account the weights considered by MERT for the in-domain development set. The idea behind such sampling is to perform a Monte Carlo-like sampling of the model parameters.

A last consideration when attempting to implement the Equation 11 is that the first part of the integral, the product over all samples in the adaptation set, cannot be computed with typical state-of-the-art phrase-based SMT systems, since e.g. out-of-vocabulary words may imply that the SMT model is unable to explain a certain bilingual sentence completely. Hence, instead of computing

$$\prod_{\forall a \in A} \frac{\exp \sum_k \theta_k \ f_k(\boldsymbol{x}_a, \boldsymbol{y}_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \ f_k(\boldsymbol{x}_a, \boldsymbol{y}')} \tag{12}$$

we will need to compute

$$\prod_{\forall a \in A} \frac{\exp \sum_k \theta_k \ f_k(\boldsymbol{x}_a, \boldsymbol{y}^*_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \ f_k(\boldsymbol{x}_a, \boldsymbol{y}')} \tag{13}$$

where $\boldsymbol{y}^*$ represents the best hypothesis the search algorithm is able to produce, according to a given translation quality measure. Since BLEU is not well defined at the level of sentence because it implements a geometrical average which can be zero, we will be using TER for this purpose.

5.3 Experimental Results

We conducted adaptation experiments by using the SMT system trained on Europarl as a baseline system and translated the Xerox and EU test sets. Then, increasing the number of adaptation samples made available to the system was considered, starting from 10 up to 140. These adaptation samples were drawn from the respective training corpus, i.e. when translating the Xerox test set, the adaptation samples were drawn from the Xerox training corpus. In order to provide robustness to the results presented here, 15 random samplings for each size of the adaptation subset were drawn. These adaptation data were used either for weight estimation via MERT, or as adaptation set for our BA technique. Results can be seen in Figures 1 and 2. It is important to remember that the higher the BLEU score the better, as opposed to TER, where lower scores imply better translation quality.

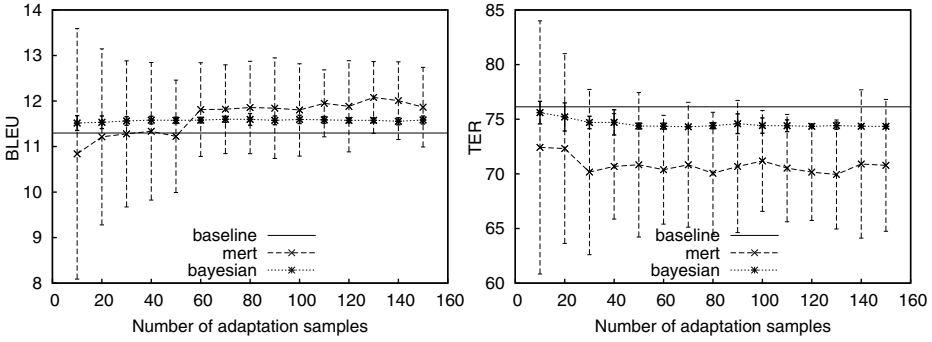

Fig. 1. Performance of baseline and both adaptation techniques when increasing the number of adaptation samples. Translation quality is measured with BLEU and TER for the Xerox test data. 95% confidence intervals are shown.

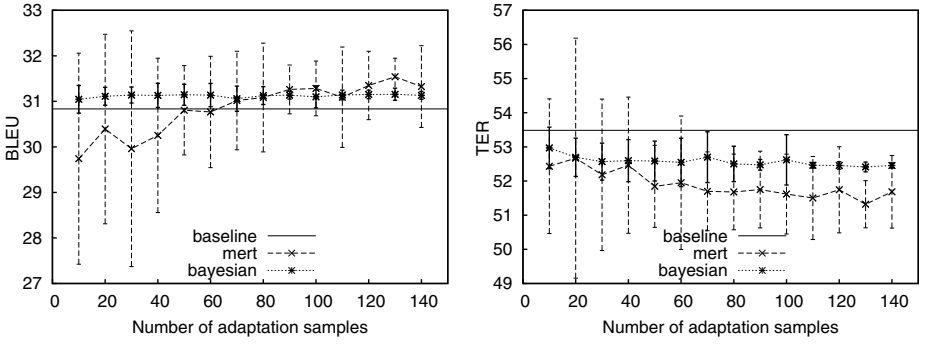

Fig. 2. Performance of baseline and both adaptation techniques when increasing the number of adaptation samples. Translation quality is measured with BLEU and TER for the EU test data. 95% confidence intervals are shown.

As the figures show, the translation quality produced by the system with $\boldsymbol{\Lambda}$ adjusted by means of MERT turns very unstable, and the confidence intervals get very big. In average, such system is able to improve the baseline, but at the risk of producing very bad quality translations. This is not an acceptable behaviour for a system that is set on-line for translating. Furthermore, the computational cost of running the MERT algorithm, even for small amounts of adaptation data, is prohibitive whenever the system is required to produce translations in real-time environments, in which the user awaits for a translation to be produced almost immediately. In contrast, the BA technique is able to yield improvements over the baseline translation quality even when very small amounts of adaptation data are available, with a much more predictable behaviour: while the confidence intervals have a range of about 7 points for BLEU and even 23 points for TER, BA is able to reduce the intervals to less than a single point in almost every case. Although estimating $\boldsymbol{\Lambda}$ only on the adaptation set seems to perform *on average* better than

BA, this comes at the risk of producing much worse translations. Moreover, the formula described in 11 can easily be incorporated into the decoder, without any significant increase in computational complexity.

6 Conclusions and Future Work

The results presented in the previous section show that the BA technique implemented is able to provide consistent improvements over the baseline, although these are not very big, even when a very small amount of adaptation data is available. Precisely in this scenario is in which a true adaptation technique is to be applied: if enough adaptation data is available, then the best "adapted" system is the system trained only on the adaptation data. Hence, when the amount of adaptation data available increases, MERT is able to yield better results. However, it must also be noted that MERT heavily depends on the data provided, as the confidence intervals show, and this can lead to unexpectedly high or low translation quality without being able to know the behaviour in advance.

Nevertheless, there are several details that must still be taken care of, and that we plan to address in future work. First, if we look at Equation 11, it seems very obvious that the first and the second component of the integral, i.e. the probability of the adaptation data and the prior over the model parameters, are clearly in very different numeric ranges. This has as effect that the probability of the adaptation sample may have less discriminative power than the prior, and this, in turn, may be the reason why the results presented are so stable, but do not yield very big improvements. We plan to address this in future work by introducing weighting coefficients to compensate for this. Such coefficients might need to be trained, but most likely only once, independently of the corpus used.

The way in which the weight sampling is done is also bound to have an important impact on the final results. We also plan to address it in future work.

The derivation presented here can be quite 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 the quality of the translations produced, since the amount of parameters to be adapted is much higher.

Acknowledgements

Work partially supported by the EC (FEDER/FSE), the Spanish Government (MICINN, MITyC, "Plan E") under MIPRCV "Consolider Ingenio 2010" (CSD2007-00018), erudito.com (TSI-020110-2009-439) and iTrans2 (TIN2009-14511), and by the Generalitat Valenciana (Prometeo/2009/014). The authors would like to thank the anonimous reviewers for their constructive comments.

References

1. Brown, P., Pietra, S.D., Pietra, V.D., Mercer, R.: The mathematics of machine translation. In: Computational Linguistics, vol. 19, pp. 263–311 (June 1993)
2. Papineni, K., Roukos, S., Ward, T.: Maximum likelihood and discriminative training of direct translation models. In: Proc. of ICASSP, pp. 189–192 (1998)

3. Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: Proc. of the ACL 2002, pp. 295–302 (2002)
4. Heigold, G., Rybach, D., Schlüter, R., Ney, H.: Investigations on convex optimization using log-linear hmms for digit string recognition. In: Interspeech, Brighton, U.K., September 2009, pp. 216–219 (2009)
5. Tahir, M.A., Heigold, G., Plahl, C., Schlueter, R., Ney, H.: Log-linear framework for linear feature transformations in speech recognition. In: IEEE Automatic Speech Recognition and Understanding Workshop, Merano, Italy (December 2009)
6. Lagarda, A., Juan, A.: Topic detection and classification techniques. In: WP4 deliverable, TransType2 (2003)
7. Nepveu, L., Lapalme, G., Langlais, P., Foster, G.: Adaptive language and translation models for interactive machine translation. In: Proc. of EMNLP (2004)
8. Kuhn, R., Mori, R.D.: A cache-based natural language model for speech recognition. IEEE Transactions on PAMI 12(6), 570–583 (1990)
9. Koehn, P., Schroeder, J.: Experiments in domain adaptation for statistical machine translation. In: Proc. of ACL WMT (2007)
10. Bertoldi, N., Federico, M.: Domain adaptation in statistical machine translation with monolingual resources. In: Proc. of EACL WMT (2009)
11. Civera, J., Juan, A.: Domain adaptation in statistical machine translation with mixture modelling. In: Proc. of ACL WMT (2007)
12. Zhao, B., Eck, M., Vogel, S.: Language model adaptation for statistical machine translation with structured query models. In: Proc. of CoLing (2004)
13. Sanchis-Trilles, G., Cettolo, M., Bertoldi, N., Federico, M.: Online Language Model Adaptation for Spoken Dialog Translation. In: Proc. of IWSLT, Tokyo (2009)
14. Zhang, H., Quirk, C., Moore, R.C., Gildea, D.: Bayesian learning of non-compositional phrases with synchronous parsing. In: Proceedings of ACL 2008: HLT. Association for Computational Linguistics, June 2008, pp. 97–105 (2008)
15. Zens, R., Och, F., Ney, H.: Phrase-based statistical machine translation. In: Jarke, M., Koehler, J., Lakemeyer, G. (eds.) KI 2002. LNCS (LNAI), vol. 2479, pp. 18–32. Springer, Heidelberg (2002)
16. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proc. HLT/NAACL 2003, pp. 48–54 (2003)
17. Och, F.: Minimum error rate training for statistical machine translation. In: Proc. of Annual Meeting of the ACL (July 2003)
18. Papineni, K., Kishore, A., Roukos, S., Ward, T., Zhu, W.J.: Bleu: A method for automatic evaluation of machine translation. In: Technical Report RC22176, W0109-022 (2001)
19. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proc. of AMTA 2006 (2006)
20. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT Summit (2005)
21. Koehn, P., Monz, C. (eds.): Proc. on the Workshop on SMT. Association for Computational Linguistics (June 2006)
22. Esteban, J., Lorenzo, J., Valderrábanos, A., Lapalme, G.: Transtype2 - an innovative computer-assisted translation system. In: Proc. of 42nd ACL, Barcelona, Spain, July 2004, pp. 94–97 (2004)
23. Khadivi, S., Goute, C.: Tools for corpus alignment and evaluation of the alignments (deliverable d4.9). In: Technical Report, TransType2, IST-2001-32091 (2003)
24. Koehn, P., et al.: Moses: Open source toolkit for statistical machine translation. In: Proc. of ACL Demo and Poster Sessions, Czech Republic, Prague, pp. 177–180 (2007)