Online Language Model adaptation via N-gram Mixtures for Statistical Machine Translation

Germán Sanchis-Trilles
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Valencia, Spain
gsanchis@dsic.upv.es

Mauro Cettolo
FBK
Fondazione Bruno Kessler
Trento, Italy
cettolo@fbk.eu

Abstract

The problem of language model adaptation in statistical machine translation is considered. A mixture of language models is employed, which is obtained by clustering the bilingual training data. Unsupervised clustering is guided by either the development or the test set. Different mixture weight estimation schemes are proposed and compared, at the level of either single or all source sentences. Experimental results show that, by training different specific language models weighted according to the actual input instead of using a single target language model, translation quality is improved, as measured by BLEU and TER.

1 Introduction

The grounds of modern Statistical Machine Translation (SMT), a pattern recognition approach to machine translation, were established in (Brown et al., 1993), where the problem of machine translation 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 maximizes the posterior probability is to be found. Such a statement can be formalized as:

$$\begin{aligned}\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f}) \\ &= \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{f}|\mathbf{e}) \cdot \Pr(\mathbf{e}) \qquad (1)\end{aligned}$$

where $\Pr(\mathbf{f}|\mathbf{e})$ stands for the translation probability and $\Pr(\mathbf{e})$ accounts for penalizing ill-formed sentences of the target language.

More recently, a direct modeling 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 model, where

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

and the decision rule is given by the expression

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

where $h_k(\mathbf{f}, \mathbf{e})$ is a score function representing an important feature for the translation of $\mathbf{f}$ into $\mathbf{e}$, for example the target language model $p(\mathbf{e})$, K is the number of models (or features) and λ_k are the weights of the log-linear combination. Typically, the weights λ_k are optimized during the tuning stage with the use of a development set.

In this paper, we deal with the problem of adaptation of SMT models. Specifically, we focus on augmenting the Language Model (LM) component by introducing parameters that are adapted dynamically to the input text. With this purpose, the LM is implemented as a mixture of specialized sub-LMs, which are conveniently estimated through some bilingual clustering of the training data and then combined following different weighting schemes. The work described here represents an important extension of what is presented in (Sanchis-Trilles et al., 2009): in fact, there the methods were tested on a small task like IWSLT; on the contrary, here the approach is assessed on the medium-sized Europarl task. Moreover, the clustering of training data does not exploit any supervised annotation of texts.

The paper is organized as follows. Section 2 briefly lists other papers dealing related issues.

Our adaptation procedure is described in Section 3, together with the different clustering techniques and weighting schemes we have investigated. In Section 4 the experimental setup is described and results provided and commented. Possible extensions of the present work are depicted in Section 5; some final remarks ends the paper.

2 Related Work

LM adaptation has been deeply explored since at least mid 90s in the ambit of speech recognition (De Mori and Federico, 1999; Bellagarda, 2001). Nowadays, also in the SMT community the interest for adaptation is continuously growing. One of the first approaches was proposed by (Lagarda and Juan, 2003), in which the translation model (TM) is implemented as an unsupervised multinomial mixture of TMs and each component is supposed to concentrate most of its probability mass on a certain topic. Slightly later, (Nepveu et al., 2004) applied other adaptation techniques to interactive MT, following the ideas in (Kuhn and De Mori, 1990) and adding cache LMs and cache 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, the alignments being used to extract phrases.

A work that resembles the one presented here is (Zhao et al., 2004), where each source sentence was used to build a query and retrieve similar sentences from a larger corpus. Then, a specific LM was trained and interpolated with a generic LM. This combination was used to translate the original sentence. In (Lü et al., 2007), each sentence was used to select similar data within the same corpus by means of TF-IDF, and then prepare specific LMs and TMs ready to be interpolated. In (Yamamoto and Sumita, 2007), the bilingual corpus is clustered so as to minimize the entropy of each subset, and then independent LMs and TMs are trained from these smaller bilingual corpora, which are in turn recombined in translation time by performing domain prediction. Differently, in our work the final combination of target LMs is obtained by re-using the weights estimated by maximizing the probability of generating the source sentence by means of the linear interpolation of source sub-LMs.

3 Language Model Adaptation

In this paper, we focus on the problem of LM adaptation. Specifically, one of the features described in eq. 3 may be

$$h(\mathbf{e}, \mathbf{f}) = \log p(\mathbf{e})$$

which provides the log score of the target LM. Typically, $p(\mathbf{e})$ is given by a single LM; this configuration will represent our `baseline`. However, that distribution can be expressed also as a linear interpolation (mixture) of LMs:

$$p(\mathbf{e}) = \sum_{i=1}^{M} w_i p_i(\mathbf{e})$$

where p_i's are target LMs built on clusters which the training data are split in. Our aim is to adapt the interpolation of LMs by tuning the weights on the actual input. With the help of Figure 1, the basic adaptation procedure is described in the following.

Let us assume that the parallel training data have been partitioned into a set of `M` bilingual clusters, according to some criterion. On each cluster, language specific LMs are estimated, which are then organized into two language specific mixture models, one modeling the source language, the other the target language. So far, operations can be performed off-line. Now let us consider a source text or sentence to be translated. Before translation, the input is used to estimate optimal weights of the source language mixture through Expectation-Maximization (EM). The resulting weights are then transferred to the target language mixture, which is finally used as LM feature function by the SMT system. The rationale behind the "weight transfer trick" is that clustering likely generates sub-models specialized in terms of contents rather than linguistic structure and then the convenience to weight more or less a given sub-model is expected to be shared by both source and target sides.

3.1 Clustering

It should be clear that the fundamental intermediate step of our approach is the clustering of bilingual training data. The elements of each cluster are sentences. Hence, the goal of this stage is to group together sentences which are similar each other from the lexical point of view. Unless differently specified, the clustering is performed by

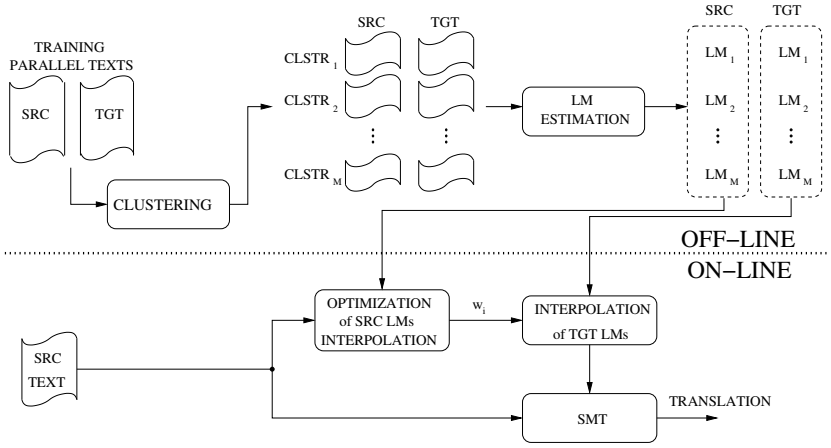

Figure 1: *Basic procedure for LM adaptation.*

- representing each sentence pair as a bag of both source and target words;
- setting the number of clusters to 4, since a preliminary investigation revealed this number as begin able to generate clusters quite specialized and not too sparse;
- means of the CLUTO[1] package. The chosen setup includes the k-way partitioning scheme and the cosine distance as similarity function between sentences.

On both source and target sides, in addition to the 4 LMs trained on each cluster and for smoothing purposes, the LM built on the whole training data has also been considered.

In the following subsections, three different clustering schemes are described.

3.1.1 Direct clustering

As a first approach, we investigated the direct clustering of the bilingual training data by means of CLUTO.

3.1.2 Development-induced clustering

Although the direct clustering of the training data is the most straightforward choice, it might not be the best choice, since by definition the goal of any adaptation procedure is to cover possible mismatches between training and development/test conditions. With this in mind, we propose to induce the clustering of the training data from the clusters computed on the development set. The scheme is shown in Figure 2 and is summarized in the following algorithm:

1. clustering the bilingual development text
2. estimate source and target LMs for each cluster from step (1)
3. partition training data by classifying each sentence pair according to eq. 4 (see below)

In step (3), each bilingual training sentence n is assigned to the cluster $\hat{m}$ by the rule:

$$\hat{m} = \underset{m}{\text{argmax}}\, cos(\mathbf{t}_n^{src}, \mathbf{d}_m^{src}) + cos(\mathbf{t}_n^{tgt}, \mathbf{d}_m^{tgt}) \quad (4)$$

where $\mathbf{t}$ and $\mathbf{d}$ are vectors of M (the number of clusters) weights and the cosine between two vectors is defined as $cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||\,||\mathbf{y}||}$, with $\cdot$ being the dot product and $||\ ||$ being the 2-norm. In particular, $\mathbf{t}_n^{src}$ is such to maximize the probability of the linear interpolation of source LMs estimated in step (2) on the source sentence n of the training text; the maximization is performed by an EM step. $\mathbf{t}_n^{tgt}$ is the twin of $\mathbf{t}_n^{src}$ for the target side. $\mathbf{d}_m^{src}$ ($\mathbf{d}_m^{tgt}$) is the vector of weights which maximize the probability of again the source (target) LMs of step (2) but on the whole source (target) side of cluster m of the partitioning of the development set.

The intuitive explanation of eq. 4 relies on the meaning of components of vectors $\mathbf{t}$ and $\mathbf{d}$. Let us start by the fact that in some sense a LM trained from a specific cluster is a compact representation of the sentences in that cluster; hence, the optimization of LM weights on a text provides, through each single weight, a measure of the similarity of that text with a specific LM, that is a specific cluster. Vectors $\mathbf{t}$ and $\mathbf{d}$ can then be considered as "fingerprints" of each training sentence and development cluster, respectively. The $cos()$ operation on them is then applied to compute the similarity of training sentences with each cluster m.

[1] Available from http://glaros.dtc.umn.edu/gkhome/views/cluto

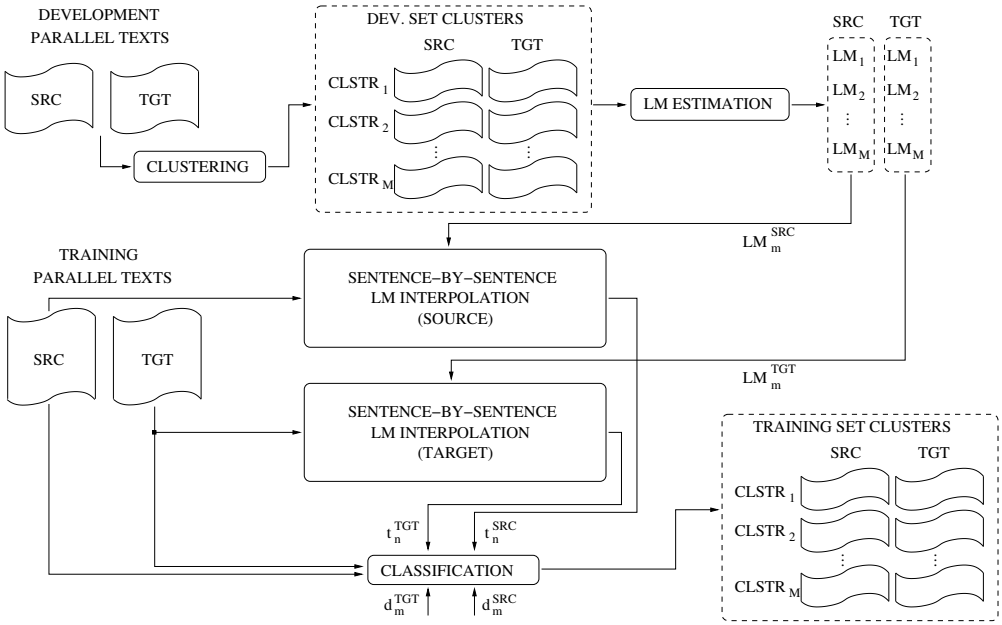

Figure 2: *Procedure for obtaining development-induced clustering of the training data.*

3.1.3 Test-induced clustering

For inducing the clustering of the bitext training data it is possible to use the test set instead of the development set. In this case the target side is not available, then the clustering is performed only on the source data, and the classification rule of eq. 4 is modified accordingly:

$$\hat{m} = \underset{m}{\text{argmax}}\, cos(\mathbf{t}_n^{src}, \mathbf{d}_m^{src}) \qquad (5)$$

Note that even if eq. 5 relies only on the source side, it is used to classify both sides of each sentence n of the training data.

The idea behind performing a test-induced clustering is that of taking profit of the information available in the actual text to be translated. Nevertheless, the possible benefits of using such information may not be completely reliable, since only the source side is available and the clustering is instead induced on bilingual data.

3.2 Weight optimization

Once training text has been clustered and LMs have been estimated for each cluster, weights are needed for performing the interpolation. We investigated three different approaches, each one with a different degree of granularity but with the common attempt of exploiting the actual input.

3.2.1 Set specific weights

LM-interpolation weights are estimated on the source side of the complete test set. This approach, which is the most straightforward, has nevertheless an important drawback: the weights estimated are those that well model the whole test set on average, without considering possibly significant differences between specific sentences. Hence, the potential benefit of using several LMs may fade.

3.2.2 Sentence specific weights

In this case, one specific set of weights is estimated for each sentence of the test set. By doing so, we expect that the effect of splitting the training corpus into several subsets yields better results, since the EM procedure is allowed complete freedom in assigning the LM weights. However, weights computed in such a manner may be less reliable, since the estimation is performed on few data (one single sentence).

3.2.3 Two-steps weight estimation

This approach merges the previous two in the attempt of keeping their advantages and overcoming the drawbacks. Once sentence specific weights have been computed, each (source) test sentence is assigned to the specific cluster corresponding to the most weighted LM. This being done, one set of weights can be re-estimated for each one of the test clusters obtained in this way. This approach has the intuitive benefit of mirroring the clustering of the training data into the test set, while still avoiding the possible data sparseness issue that can affect the sentence specific weight estimation. This procedure is illustrated in Figure 3.

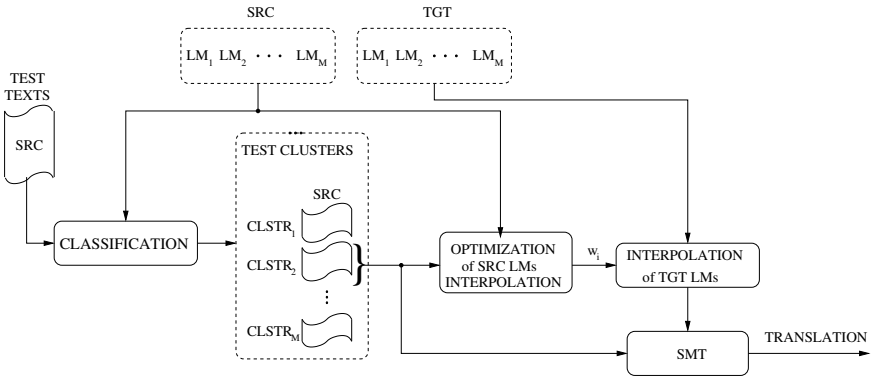

Figure 3: *Two-steps weight estimation technique.*

Table 1: Europarl corpus statistics. Average sentence length is always between 20 and 30 words. OoV stands for "Out of Vocabulary" words, Dev. for Development, K/M for thousands/millions.

		De	En	Es	En	Fr	En
Training	Sentences	751K		731K		688K	
	Run. words	15.3M	16.1M	15.7M	15.2M	15.6M	13.8M
	Voc.	195K	66K	103K	64K	80K	62K
Dev.	Sentences	2000		2000		2000	
	Run. words	55K	59K	61K	59K	67K	59K
	OoV	432	125	208	127	144	138
Test	Sentences	2000		2000		2000	
	Run. words	54K	58K	60K	58K	66K	58K
	OoV	377	127	207	125	139	133

4 Experiments

4.1 Corpora

Experiments were conducted on the Europarl corpus (Koehn, 2005), in the setup established in the Workshop on Statistical Machine Translation of the NAACL 2006 (Koehn and Monz, 2006).

The Europarl corpus consists of the transcriptions of European Parliament speeches and includes versions in eleven European languages. Here, we will focus on the German–English, Spanish–English and French–English tasks, the same language pairs selected for the cited workshop. Although we tested our systems on both translation directions, in this paper we will only report experiments having English as source language for the sake of brevity, given that the behavior in the opposite direction was similar. The corpus is divided into three separate sets, for training, development and testing purposes, respectively. Statistics are provided in Table 1.

4.2 Baseline system

The baseline system is built upon the open-source MT toolkit Moses (Koehn et al., 2007)[2] in its default setup. Following standard practice, the weights of the log-linear combination (eq. 3) are optimized by means of the Minimum Error Rate Training (MERT) procedure (Och, 2003). MERT was only performed for the baseline system, and its weights were re-used for all other systems. Although there could be reasons for re-running MERT when the LM changes, we did not do so in order to better isolate the effects of including different LMs into the SMT system.

As baseline LM, a 5-gram word-based LM was estimated on the target side of the training corpus, smoothed according to the improved Kneser-Ney technique (Chen and Goodman, 1999).

4.3 Results

Adaptation procedures presented in Section 3 have been experimentally assessed by performing automatic translation whose quality is measured in terms of BLEU (Papineni et al., 2001) and TER (Snover et al., 2006).

Statistical significance tests have also been computed, according to the technique described in (Koehn, 2004), with 10K bootstrap repetitions.

The three tables presented in the following collect BLEU and TER scores; in the additional column `Signif`, the result of the statistical significance tests is provided in binary terms (yes/no) by checking if the improvement (or drop) in translation quality with respect to the baseline performance is significant at a 95% confidence level. These tests have been computed and outcomes are provided for both BLEU/TER scores.

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

Table 2: Performance of the direct clustering approach.

Language pair	Weight optimization	PP	BLEU	TER	Signif BLEU/TER
En–Es	baseline	78.5	30.8	54.9	–
	sentence	71.3	30.4	54.6	yes/yes
	two-steps	71.2	30.3	54.5	yes/yes
	test set	100.1	30.3	54.5	yes/yes
En–De	baseline	141.5	19.0	67.4	–
	sentence	129.0	18.2	67.4	yes/no
	two-steps	129.7	18.1	67.4	yes/no
	test set	202.3	18.0	67.6	yes/no
En–Fr	baseline	50.0	32.9	55.3	–
	sentence	45.4	32.7	55.0	no/yes
	two-steps	45.5	32.6	54.9	yes/yes
	test set	64.5	32.5	55.0	yes/yes

Table 3: Performance of the development-induced clustering approach.

Language pair	Weight optimization	PP	BLEU	TER	Signif BLEU/TER
En–Es	baseline	78.5	30.8	54.9	–
	sentence	68.3	**31.3**	54.4	yes/yes
	two-steps	68.3	**31.3**	**54.3**	yes/yes
	test set	105.6	30.9	54.6	yes/yes
En–De	baseline	141.5	19.0	67.4	–
	Sentence	126.0	**19.2**	**66.7**	yes/yes
	two-steps	126.3	**19.2**	**66.7**	yes/yes
	test set	206.6	18.7	67.2	yes/no
En–Fr	baseline	50.0	32.9	55.3	–
	sentence	43.5	33.2	54.9	yes/yes
	two-steps	43.5	**33.3**	**54.8**	yes/yes
	test set	65.0	32.9	55.1	no/yes

Finally, the column PP shows the perplexity value of either the single LM (baseline) or the interpolation of LMs (other cases) computed on the test set references.

4.3.1 Direct clustering

Results observed by directly clustering the training data are shown in Table 2, for all the three weight optimization schemes and for all the three translation directions.

A degradation of the BLEU score is observed in any condition, while TER slightly improves for the En–Es and En–Fr pairs, especially when either the sentence-based or the two-steps estimation schemes are adopted. However, since results are not coherent for two scores, it cannot be definitely stated whether this form of LM adaptation overcomes the use of the single baseline LM.

4.3.2 Development-induced clustering

Results for the development-induced clustering are reported in Table 3. In this case, the LM adaptation does improve the baseline consistently, for both scores and significantly in almost every setup. Again, the best performing weight optimization scheme is the two-steps one, which improves the baseline in all language pairs in a statistically significant way. Performances comparable to those of two-steps optimization are obtained also with weights estimated at the single test sentence level. Again, the optimization of weights on the whole test set does not seem to be appropriate.

4.3.3 Test-induced clustering

Lastly, Table 4 collects results when the clustering of training data is induced by the test set. This kind of clustering seems not to be able to exploit the test information provided to the system; in fact, BLEU is non-differentiable from the baseline in almost every setup, while TER is improved only at a limited extent. Concerning the weight optimization, here the best choice is to perform it on the whole test set, differently from what happened in the other types of clustering. This could be originated from the fact that LMs are built on clusters induced by just the test set. For this reason, in this specific case the use of the whole test set allows an effective trade-off between the estimation of weights which are good on average on the whole test set and the sparseness of data on which the optimization is done. Nevertheless, it is worth noticing that differences in translation quality are mostly not statistically significant.

4.4 General remarks

Results in Tables 2, 3 and 4 show the different impact that the proposed clustering and weight optimization schemes for LM adaptation have on MT performance. In particular, the best scores measured in our experiments, marked in bold in Table 3, are achieved when using development-induced clustering combined with the two-steps (or sentence-based) weight optimization. With this setup, the translation quality always improves the one obtained by the baseline system. Such results, which are statistically significant and coherent throughout all language pairs and for both considered evaluation scores, prove that there is a potential benefit behind the use of n-gram mixtures in SMT.

From another viewpoint, it seems that the sentence-based interpolation technique is able to

yield the same translation quality than the two-steps weight optimization. This should indirectly prove that the input sentence alone contains sufficient information to make the interpolation procedure stable enough. In fact, average sentence length for the test sets ranges from 33 words per sentence for French, to 27 words per sentence for German, i.e. fairly long sentences. Given this experimental evidence and the fact that it is computationally cheaper, the sentence-based optimization should be the first choice in presence of quite long input sentences.

It must also be noted that, although all the subsets of the Europarl corpora belong to the same domain, they were not extracted randomly: specifically, the training corpus comprises data from year 1997 to year 2003, although the development and test data are extracted from the fourth quarter of year 2000. This fact should explain the good results obtained with the development-induced clustering, since both test and development sets belong to a very narrow time frame, in which the topics being debated in the European Parliament were likely similar. Hence, development-induced clustering may be able to make a better use of the uneven distribution of training and development/test data, since it resembles the test data, and contains bilingual information (as opposed to test-induced clustering).

The fact that test-driven clustering only relies on source-sentence information is an important drawback that cannot be ignored: preliminary investigations revealed that including both source and target information into the clustering procedure did have an important impact, which is evidenced in this case as well. Although it might seem that monolingual clustering relies on half of the information of bilingual clustering, this is even optimistic: in fact, bilingual clustering does not only take into account both source and target sides, but also the interaction between the two.

5 Future Work

Results achieved in this work reveals that the improvements that can be obtained by our LM adaptation approach greatly depend on the clustering technique employed. Since here only the surface form of single words has been used for clustering the training data, we plan to investigate alternatives, such as clustering based on n-gram or PoS-tag information.

Table 4: Performance of the test-induced clustering approach.

Language pair	Weight optimization	PP	BLEU	TER	Signif BLEU/TER
En–Es	baseline	78.5	30.8	54.9	–
	sentence	72.4	30.9	54.6	no/yes
	two-steps	72.2	30.9	54.6	no/yes
	test set	105.7	31.0	54.6	yes/yes
En–De	baseline	141.5	19.0	67.4	–
	sentence	133.7	18.9	67.3	no/no
	two-steps	133.9	18.9	67.3	no/no
	test set	204.4	18.9	67.1	no/yes
En–Fr	baseline	50.0	32.9	55.3	–
	sentence	46.6	32.8	55.2	no/no
	two-steps	46.4	32.8	55.3	no/no
	test set	65.2	33.0	55.2	no/no

Furthermore, another interesting issue left out from this paper is supervised clustering. In fact, detailed supervision is typically available only for quite small linguistic resources; on the other side, large quantity of texts can be provided with coarse - and even not fully reliable - labels about the topic contents, like the xml documents made available by Google News. Then, when large sized tasks are involved, a research issue is that of how to exploit such kind of information for making more effective the clustering.

Another issue which deserves an investigation regards the interpolation of target LMs by re-using weights estimated for the optimal interpolation of source LMs. In fact, although it appears as a reasonable choice, it could happen that the likelihood on the target side is maximized with different weights than those which ensures the maximum likelihood on the source side. A source-to-target weight map could be learnt from a parallel development/training set.

6 Conclusions

This paper has presented a technique for adapting the LM of SMT systems to the actual input. The assumption is that the LM is provided as a linear interpolation of sub-LMs, each estimated on a specific portion of the training data. The interpolation weights are then estimated dynamically on the text to be translated via a maximum likelihood EM-based procedure.

Different methods for clustering training data in an unsupervised manner and different schemes for estimating the interpolation weights have been experimentally tested on three language pairs of the Europarl task. Results have showed that (i) the

clustering induced by exploiting both sides (source and target) of the development set and (ii) the estimation of weights at the sentence level or with the two-steps approach yield consistent improvements in translation quality over the reference baseline.

Acknowledgments

This work was supported by the EuroMatrix-Plus project (IST-231720), which is funded by the EC under the Seventh Framework Programme for Research and Technological Development, by the Spanish MICINN iTrans2 (TIN2009-14511) project, the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Spanish MEC under scholarship AP2005-4023.

References

Bellagarda, J. R. 2001. An overview of statistical language model adaptation. In *Proc. of ISCA Workshop on Adaptation Methods for Speech Recognition*, pp. 165–174, Sophia-Antipolis, France.

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

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Chen, S. F. and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393.

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

De Mori, R. and M. Federico. 1999. Language model adaptation. In Pointing, K., editor, *New Advances and Trends in Speech Recognition and Coding*, NATO ASI Series F, pp. 280–301. Springer Verlag, Germany.

Koehn, P. and C. Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proc of NAACL WSMT*, pp. 102–121, New York City.

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

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

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*, pp. 388–395.

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

Lagarda, A. and A. Juan. 2003. Topic detection and classification techniques. In *WP4 deliverable*, TransType2.

Lü, Y., J. Huang, and Q. Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proc. of EMNLP*.

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.J. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL*, pp. 295–302.

Och, F.J. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*, pp. 160–167, Sapporo, Japan.

Papineni, K., S. Roukos, and T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. of ICASSP*, pp. 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, Japan.

Snover, M., B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*.

Yamamoto, H. and E. Sumita. 2007. Bilingual cluster based models for statistical machine translation. In *Proc. of EMNLP*, Prague, Czech Republic.

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