Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component-Level Mixture Modelling

Pratyush Banerjee, Sudip Kumar Naskar, Johann Roturier[1], Andy Way[2*], Josef van Genabith
CNGL, School of Computing, Dublin City University, Dublin, Ireland
{pbanerjee,snaskar,josef}@computing.dcu.ie
[1] Symantec Limited, Dublin, Ireland
johann_roturier@symantec.com
[2] Applied Language Solutions, Delph, UK
andy.way@appliedlanguage.com

Abstract

This paper reports experiments on adapting components of a Statistical Machine Translation (SMT) system for the task of translating online user-generated forum data from Symantec. Such data is monolingual, and differs from available bitext MT training resources in a number of important respects. For this reason, adaptation techniques are important to achieve optimal results. We investigate the use of mixture modelling to adapt our models for this specific task. Individual models, created from different in-domain and out-of-domain data sources, are combined using linear and log-linear weighting methods for the different components of an SMT system. The results show a more profound effect of language model adaptation over translation model adaptation with respect to translation quality. Surprisingly, linear combination outperforms log-linear combination of the models. The best adapted systems provide a statistically significant improvement of 1.78 absolute BLEU points (6.85% relative) and 2.73 absolute BLEU points (8.05% relative) over the baseline system for English–German and English–French, respectively.

1 Introduction

In recent years, Statistical Machine Translation (SMT) technology has been used in many online applications, concentrating on professionally edited enterprise quality online content. At the same time, very little research has gone into adapting SMT technology to the translation of user-generated content on the web. While translation of online chats (Flournoy and Callison-Burch, 2000) has received some attention, there is surprisingly little work on translation of online user forum data, despite growing interest in the area (Flournoy and Rueppel, 2010). In this paper we describe our efforts in building a system to address this particular application area. Our experiments are conducted on data collected from online forums on Symantec Security tools and services.[1] For a multinational company like Symantec, the primary motivation behind translation of user forum data is to enable access across language barriers to information in the forums. Forum posts are rich in information about issues and problems with tools and services provided by the company, and often provide solutions to problems even before traditional customer-care help lines are even aware of them.

The major challenge in developing MT systems for user forum data concerns the lack of proper parallel training material. Forum data is monolingual and hence cannot be used directly to train SMT systems. We use parallel training data in the form of Symantec Enterprise Translation Memories (TMs) from different product and service domains to train the SMT models. As an auxiliary source, we also used portions of the Europarl dataset[2] (Koehn, 2005), selected according to their similarity with the forum data (Section 3.2), to supplement the TM-based training data. Symantec TM data, being a part of enterprise documentation, is professionally

[*] Work done while at CNGL, School of Computing, DCU

[1] http://community.norton.com/

[2] http://www.statmt.org/europarl/

edited and by and large conforms to the Symantec controlled language guidelines, and is significantly different in nature from the user forum data, which is loosely moderated and does not use controlled language at all. In contrast Europarl data is out-of-domain with respect to the forum data. The differences between available training and test datasets necessitate the use of adaptation techniques for optimal translation. We use mixture model adaptation (Foster and Kuhn, 2007), creating individual models from different sources of data and combining them using different weights. Monolingual forum posts were used for language modelling along with the target side of the TM training data. A system trained only on the Symantec TM and forum data serves as the baseline system. All our experiments are conducted on the English-German (En–De) and English-French (En–Fr) language pairs with a special emphasis on translation from English. For the sake of completeness however, we report translation scores for both directions here.

Apart from using models created from concatenation of in-domain (Symantec TM) and out-of-domain (Europarl) datasets, we used linear and log-linear combination frameworks to combine individual models. Both translation models and language models were separately combined using the two methods and the effect of the adaptation was measured on the translation output using established automatic evaluation metrics. Our experiments reveal that for the current task, in terms of translation quality, language model adaptation is more effective than translation model adaptation and linear combination performs slightly better than the log-linear setting.

The remainder of this paper is organized as follows: Section 2 briefly describes related work relevant to the context. Section 3 reports the tools and algorithms used along with a description of the datasets used. Section 4 focuses on the mixture modelling experiments and how weights are learnt in different settings. Section 5 presents the experiments and analysis of results, followed by conclusions and future work in Section 6.

2 Related Work

Mixture Modelling (Hastie et al., 2001), a well-established technique for combining multiple models, has been extensively used for language model adaptation, especially in speech recognition. Iyer and Ostendorf (1996) use this technique to capture topic dependencies of words across sentences within language models. Cache-based language models (Kuhn and De Mori, 1990) and dynamic adaptation of language models (Kneser and Steinbiss, 1993) for speech recognition successfully use this technique for sub-model combinations.

Langlais (2002) introduced the concept of domain adaptation in SMT by integrating domain-specific lexicons in the translation model, resulting in significant improvement in Word Error Rate. Eck et al. (2004) utilized information retrieval theories to propose a language model adaptation technique in SMT. Hildebrand (2005) utilized this approach to select similar sentences from available training data to adapt translation models, which improved translation performance with respect to a baseline system. Wu et al. (2008) used a combination of in-domain bilingual dictionaries and monolingual data to perform domain adaptation for SMT in a setting where in-domain bilingual data was absent. Integrating an in-domain language model with an out-of-domain one using log-linear features of a phrase-based SMT system is reported by Koehn and Schroeder (2007). Foster and Kuhn (2007) used mixture modelling to combine multiple models trained on different sources and learn mixture weights based on distance of the test set from the training data. Civera and Juan (2007) further suggested a mixture adaptation approach to word alignment, generating domain-specific Viterbi alignments to feed a state-of-the-art phrase-based SMT system.

Our work follows the line of research presented in Foster and Kuhn (2007) using mixture modelling and linear/log-linear combination frameworks, but differs in terms of the test set and development sets used for tuning and evaluation. While Foster and Kuhn (2007) used test and development sets which were essentially a combination of data from different training genres, in our case test data (user forum) are inherently different from the training data. Our methods of estimating the linear weights for language and translation models are also different to the ones proposed in Foster and Kuhn (2007). As part of our experiments, we also resort to selecting portions of relevant bitext from out-of-domain corpora to augment available training data as described in Hildebrand et al. (2005). However, our work is different from their approach in the use of language model perplexity as an indicator of relevance of the selected data. Furthermore, due to the differences between the training and target datasets, we selected additional data in terms of its relevance to the target domain instead of the training domain.

3 Datasets, Pre-processing and Tools

3.1 Symantec Datasets

Our primary training data consists of En–De and En–Fr bilingual datasets in the form of Symantec TMs. Monolingual Symantec forum posts in all three languages served as language modelling data. As the purpose of our experiments is to translate forum posts, the data for the development and the test sets were randomly selected from the monolingual English forum data. After being translated using Google Translate,[3] these datasets were manually post-edited by professional translators following guidelines[4] for achieving 'good enough quality', in order to generate bilingual development (dev) and test sets. The selected test data was excluded from the English forum data used to create language models in the experiments.

Data Set	En–De	En–Fr
Symantec TM	638600	567641
Europarl	705676 (∼40%)	414667 (∼23%)
Development Set	500	500
Test Set	612	612
English Forum	1069464	
German Forum	25169	
French Forum	22932	

Table 1: Number of Sentences for bilingual training, development and test and monolingual forum data sets

Apart from the Symantec datasets, we used portions of the Europarl dataset (Section 3.2) to supplement the training data. Table 1 presents the numbers of sentences for each of the resources used in our experiments.

3.2 Extracting Relevant Data from Europarl

Given that we needed additional resources to improve translation coverage, we selected the Europarl dataset, containing parallel sentences of the proceedings of the European Parliament. However, Europarl data is clearly out-of-domain given our specific task, but much larger in size than the Symantec TM data. For this reason, we decided to select only a portion of the Europarl data in order to balance the amount of in-domain and out-of-domain data. In order to achieve this, the entire set of Europarl sentences were ranked using the sentence-level perplexity scores with respect to language models created on the monolingual forum data. Only a portion of the ranked list with scores lower than a manually chosen threshold (perplexity value of 350) were selected for our experiments. Lower perplexity scores of the included sentences indicate a closer fit (hence higher relevance) to the forum data. This technique enables us to select the most 'forum-like' sentences from Europarl. The number of selected Europarl sentences, as reported in Table 1, constitute about 40% and 23% of the total Europarl sentences for En–De and En–Fr language pairs respectively.

3.3 Preprocessing and Data Cleanup

Re: No right click scan No i copyed the file in stead of creating shortcut,LOL I did it with the shortcut and it works just fine, :) Thanks 2008-10-19T23:14:38+00:00
Re: Norton AntiBot - possible vulnerability? This has been answered on a separate thread: http://community.norton.com/norton/board/message?board.id=other&thread.id=2533&jump=true I am locking this thread now;
avibuzz wrote:Did not work I went the highkey below and could not find anything...HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run What did you find when you click on that key?

Table 2: Few examples of the untranslatable tokens in forum posts

The Symantec TM datasets and the forum posts contain many tokens unsuitable for translation including: URLs, file paths and file names, Windows registry entries, date and time stamps, XML and HTML tags, smilies, text-speak and garbage characters. Table 2 shows a few examples of forum posts containing such tokens, which we handled in the pre-processing steps using regular expressions to replace them with unique place hold-

[3] http://translate.google.com/

[4] http://www.cngl.ie/node/2542

ers. In the post-processing step, the place holders were replaced with the actual tokens, except for the smilies, text-speak and garbage characters. For entries with multiple tokens of a single type, tokens were replaced in the translation in the same order as they appeared in the source. Furthermore, prior to training, all datasets involved in the experiments were subjected to deduplication, lower casing and tokenization.

3.4 Translation and Language Models

For our translation experiments we used OpenMaTrEx (Stroppa and Way, 2006), an open source SMT system which provides a wrapper around the standard log-linear phrase-based SMT system Moses (Koehn et al., 2007). Word alignment was performed using Giza++ (Och and Ney, 2003). The phrase and the reordering tables were built on the word alignments using the Moses training script. The feature weights for the log-linear combination of the feature functions were tuned using Minimum Error Rate Training (MERT) (Och, 2003) on the devset in terms of BLEU (Papineni et al., 2002).

We used 5-gram language models in all our experiments created using the IRSTLM (Federico et al., 2008) language modelling toolkit using Modified Kneser-Ney smoothing (Kneser and Ney, 1995). Learning linear mixture weights for combining multiple language models with respect to the development set was performed using the IRSTLM language model interpolation tools. Results of translations in every phase of our experiments were evaluated using BLEU and NIST (Doddington, 2002) scores.

4 Mixture Adaptation

In the experiments reported in this paper, mixture adaptation is involved in creating individual models from separate data sources, learning mixture weights for each model and finally using the weighted mixture of models to translate the forum data test set sentences. The models were combined using linear and log-linear combination frameworks to compare the effect of the combination techniques on translation. This section details the different aspects of the mixture adaptation.

4.1 Model Combination using Linear Weights

Individual translation or language models were linearly interpolated using the formula in (1):

$$p(x|h) = \sum_{s} \lambda_s p_s(x|h) \qquad (1)$$

where $p(x|h)$ is the language model probability or the translation model probability, $p_s(x|h)$ is the particular model trained on the training resource s, and λ_s is the corresponding weight of the particular resource, all of which sum up to 1. For a linear-interpolated model, the resource weights are global weights unlike the model feature weights mentioned in Sub-section 3.4. Hence, during tuning, the linear mixture weights do not directly participate in the log-linear combination of model features.

In order to set the linear mixture weights for language models, we used the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to estimate optimal weights of the individual language models with respect to the target side of the devset. Initially all models are uniformly weighted and the EM algorithm iteratively optimizes the weights until a predefined convergence criterion is met. For translation models, we used a slightly different method to estimate the mixture weights for multiple phrase tables from different resources. Since the maximum phrase length for our SMT phrase-tables had been set to 7, we constructed 7-gram language models using the source side of the training data for each resource. The mixture weights of these language models were estimated on the devset, again using the EM algorithm. Finally the weights learned were used to combine different phrase tables. The weights set by the EM algorithm essentially denote the fitness of each data source with respect to the devset. Standard algorithms like MERT cannot effectively be used in estimating linear weights for the translation models as they are designed specifically for flat log-linear models (Foster and Kuhn, 2007).

The phrase tables constructed from the training data using Moses feature five sets of scores.

1. Inverse phrase translation probability: $\phi(f|e)$
2. Inverse lexical weight: $lex(f|e)$
3. Direct phrase translation probabilities: $\phi(e|f)$

4. Direct lexical weight: $lex(e|f)$
5. Phrase penalty: (always $exp(1) = 2.718$)

where f is the source phrase and e denotes the corresponding target phrase. Linearly mixing different phrase tables required combining their phrase translation probabilities and lexical weights as per equation (1) with linear mixture weights learnt using the EM algorithm. However, only the phrase pairs common to all the phrase tables were mixed; other phrase pairs were simply added to generate a single mixture-adapted phrase table.

4.2 Model Combination using Log-Linear Weights

We combine multiple models under the log-linear combination framework as described in equation (2):

$$p(x|h) = \prod_s p_s(x|h)^{\alpha_s} \qquad (2)$$

where α_s is the log-linear weight for the model $p_s(x|h)$ trained on the training resource s.

The advantage of using the log-linear mixture of models is that it easily fits into the log-linear framework that the SMT model is built upon. The mixture weights were estimated by running MERT on the devset with multiple phrase tables and language models. Since MERT directly optimizes the feature function weights for each available model, simply adding the different phrase tables and/or language models to the Moses configuration and using the multiple decoding path functionality (Koehn and Schroeder, 2007) of the decoder allowed us to estimate the log-linear mixture weights for each model. An added advantage is the fact that the weights are optimized not in terms of fitness to the target domain, but directly in terms of translation scores for the target domain. However, using multiple phrase tables and language models greatly increases the number of features to be optimized, thus reducing the chances of successful convergence of the MERT algorithm.

5 Experiments and Results

The adaptation experiments were conducted in three separate phases with different adaptation settings for the translation models. Within each phase, three different adaptation settings for language models were used. Conducting separate experiments for language and translation model adaptation allowed us to examine the effect of mixture modelling for the task at hand, as well as observing the effect of adaptation at each component level of an SMT system. The details of the baseline system and each phase are described in the following sections.

5.1 Baseline: Unadapted Model

The baseline system used in our experiments was a vanilla Moses system trained with the different Symantec datasets we had at our disposal. The translation models were trained on the Symantec TM data, and the language models were trained on the monolingual forum data along with the target side of the bilingual TM data. In order to keep the baseline model unadapted, the selected ‘forum-like’ Europarl data was deliberately excluded in training the baseline system, since using relevant out-of-domain data for training can be considered to be a type of adaptation.

5.2 Phase-1: Language Model Adaptation with Unadapted Translation Model

In this phase of experiments our primary objective was to observe the effect of mixture adaptation on the language models for the task of forum data translation. In order to keep the translation model free of any adaptation, we simply concatenated together the Symantec TM and ‘forum-like’ Europarl (TM+EP) datasets to create a single model. For language modelling, we had three distinct data sources at our disposal: the monolingual forum posts, the target side of the Symantec TM data, and the target side of the Europarl data. In this way we created the following three types of language models from the data sources and used them for translation.

1. **conc**: a language model trained on the concatenated data sets from all three sources, monolingual forum posts, target side of Symantec TMs and target side of ‘forum-like’ sub-parts of Europarl.
2. **linmix**: an adapted language model using linear mixture of weights.
3. **logmix**: an adapted language model using log-linear mixture of weights.

Table 3 reports the evaluation results for all phases of experiments. The first row gives the scores for the

	TM	LM	De-En		En–De		Fr-En		En–Fr	
			BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
bl	TM	TM+forum	*35.38*	*7.26*	*25.99*	*6.44*	*36.42*	*7.43*	*33.9*	*6.78*
phase-1	TM+EP	conc	35.44	7.32	26.84	6.69	36.81	7.49	35.74	7.18
	TM+EP	linmix	35.61	7.35	27.05	6.71	36.92	7.5	36.46	7.22
	TM+EP	logmix	35.49	7.31	26.98	6.5	36.74	7.46	35.89	7.13
phase-2	linmix	conc	35.03	7.27	26.98	6.53	36.56	7.41	35.99	7.17
	linmix	linmix	35.32	7.37	**27.77**	6.7	**37.1**	**7.49**	**36.63**	**7.23**
	linmix	logmix	**36.57**	**7.38**	27.39	6.67	36.74	7.42	34.51	7.02
phase-3	logmix	conc	34.82	7.31	27.23	**6.71**	34.88	7.32	32.65	6.91
	logmix	linmix	35.55	7.32	27.71	6.7	36.52	7.42	36.48	7.2
	logmix	logmix	34	7.18	27.03	6.44	36.39	7.36	34.87	6.94

Table 3: Evaluation results for all combinations of mixture adapted language and translation models: Baseline(bl) scores are italicized, best scores are in bold

baseline system. As is evident from the table, all the phase-1 experiments improve the evaluation scores over the baseline. Adding the Europarl data for training gives a slight improvement over the baseline, and both linear and log-linear mixture-adapted models further improve the scores. Surprisingly, the linear mixture results are slightly better than the log-linear ones. Since MERT directly optimizes the log-linear weights on the devset BLEU scores, as compared to the linear weights which were learnt by optimizing the maximum likelihood on the target side of the devset, we expected the former to provide better results in terms of BLEU. However, in the tuning phase, MERT was observed to iterate to the maximum allowable iteration limit (25) in order to complete, rather than converging automatically based on the evaluation metric criterion. This observation confirms previous findings (Chiang et al., 2009) regarding the inability of the MERT algorithm to converge on an optimal set of weights for a reasonably large number of parameters. Linear mixture adaptation caused the translation scores to improve by 1.06 absolute BLEU points (4.08% relative) for En–De and 2.56 absolute points (7.55% relative) for En–Fr over the baseline. For De-En and Fr-En the improvements were 0.23 absolute BLEU points (0.65% relative) and 0.5 absolute BLEU points (1.37% relative) respectively. When translating from English the improvements were statistically significant (with 97% and 99.8% reliability for En–De and En–Fr respectively), at the p=0.05 level using bootstrap resampling (Koehn, 2004). This is due to the fact that German and French forum data were smaller than the English corpus. When translating into English, however, the huge amount of monolingual English forum data used for language modelling seemed to reduce the effect of adaptation, resulting in smaller statistically insignificant absolute improvements.

Notably, in spite of being slightly worse than the linear-mixture scores, the log-linear scores are also better than the baseline scores, indicating the effectiveness of adaptation in the current setting. The NIST scores reported in the table also follow a similar trend to the BLEU scores, but the log-linear scores are slightly worse than the concatenated model scores. This might be due to the fact that MERT optimizes on BLEU scores rather than NIST to learn log-linear weights.

5.3 Phase-2: Linear Mixture Adaptation of Translation Models

In the second phase of our experiments, we extended mixture adaptation to the translation models, adapting the phrase tables using linear mixture weights. Two independent phrase tables were prepared from the Symantec TMs and ‘forum-like’ Europarl datasets which were linearly combined using weights learnt according to the process elaborated in Section 4.1. The combined phrase table was then used in combination with the different language models mentioned in Section 5.2.

The Phase-2 labelled rows in Table 3 show the results for this phase, which show very similar trends compared to Table 3 with the linear mixture-adapted language models, which resulted in best translation scores. The log-linear mixture-adapted language model performs better only for De-En translations. Using the concatenated language model with the adapted phrase table provides slightly higher translation scores compared to the ones reported in Section 5.2, suggesting a positive effect of phrase-table adaptation.

Linear mixture adaptation on phrase tables resulted in an improvement of 1.78 absolute BLEU points (6.85% relative) for En–De and 2.73 absolute BLEU points (8.05% relative) for En–Fr, over the baseline, which are better than the improvements reported in the previous section. Both these improvements are statistically significant with a reliability of 99.6% and 99.8% respectively. For De-En and Fr-En, the improvements are 1.19 absolute BLEU points (3.36% relative) and 0.68 absolute BLEU points (1.87% relative), respectively. Similarly for the concatenated translation model, improvements were slightly bigger when translating from English. The NIST scores followed the same trend as the BLEU scores in terms of relative variations.

5.4 Phase-3: Log-linear Mixture Adaptation of Translation Models

Finally, we combined multiple translation models using a log-linear combination and used them with three different language models, as in the first and second phases, and obtained the set of results reported in the phase-3 section of Table 3.

The scores follow the same trend as in the two previous phases, with the linear-adapted language model providing the best scores. The evaluation scores when translating from English were better compared to those in phase 1, but poorer than those in phase 2. The BLEU score improvements over the baseline for this adaptation model were 1.72 absolute (6.62% relative) points for En–De, 2.58 absolute (7.61% relative) points for En–Fr, 0.17 absolute (0.48% relative) points for De-En and 0.1 absolute (0.28% relative) points for Fr-En. As in the previous phases, the improvements are statistically significant for translations from English. The MERT algorithm is known to be unable to learn optimal weights for large parameter settings (Chiang et al., 2009). In the current scenario, two phrase tables, two reordering models and three language models resulted in a considerable number of parameters, causing the algorithm to learn sub-optimal mixture weights leading to poorer performance.

6 Conclusion and Future Work

The overall trends of the results emphasize the importance of linear mixture adaptation for both language and translation models. However, comparing the scores of different translation model adaptations against those of the language models indicates that language model adaptation was slightly more significant in improving translation quality, compared to translation model adaptation, for the task at hand. Although log-linear mixture adaptation fits well into the SMT framework, the inability of MERT to converge on optimal weights in different settings caused poor performance in terms of evaluation scores.

Here the weights for linear combination of multiple phrase tables were estimated using language models. Directly learning linear weights by optimizing translation quality in terms of the development set would be the prime direction in future. We would also like to look into alternative tuning techniques, especially ones based on the MIRA algorithm to improve the quality of log-linear mixture adaptation in large parameter settings (Chiang et al., 2009). Enhancing the translation quality further with third party forum data would also be another objective in this direction. Finally we would also like to investigate further on different ranking schemes and empirical threshold selection for selecting relevant datasets to supplement training data for improving translation quality.

Acknowledgments

This work is supported by Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. We thank the reviewers for their insightful comments. We also thank Symantec for kindly providing us with data and support.

References

Chiang, D., Knight, K. and Wang, W. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The*

2009 Annual Conference of the North American Chapter (NAACL'09) Boulder, CO. pp 218–226.

Civera, J. and Juan, A. (2007). Domain adaptation in statistical machine translation with mixture modelling. In *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation* Prague, Czech Republic. pp 177–180.

Dempster, A. P., Laird, N. M. and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society, Series B* Vol 39:1, pp 1–38.

Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second international conference on Human Language Technology Research* San Diego, CA, pp 138–145.

Eck, M., Vogel, S. and Waibel, A. 2004. Language model adaptation for statistical machine translation based on information retrieval In *Proceedings of the 4th International Conference on language resources and evaluation (LREC-2004)* Lisbon, Portugal, pp. 327–330

Flournoy, R., and Callison-Burch, C. 2000. Reconciling User Expectations and Translation Technology to Create a Useful Real-world Application In *Proceedings of the 22nd International Conference on Translating and the Computer* London.

Flournoy, R., and Rueppel, J. 2010. One technology: many solutions In *Proceedings of AMTA 2010: the Ninth conference of the Association for Machine Translation in the Americas* Denver, CO, pp. 6–12

Foster, G. and Kuhn, R. 2007. Mixture-model adaptation for SMT. in *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation* Prague, Czech Republic, pp.128–135.

Federico, M., Bertoldi, N. and Cettolo, M. 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech-2008* Brisbane, Australia, pp.1618–1621.

Hastie, T., Tibshirani, R. and Freidman, J. 2001. In *The Elements of Statistical Learning.* Springer-Verlag

Hildebrand, A.S.,Eck, M., Vogel S., and Waibel, A. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval In *10^{th} EAMT Conference: Practical Applications of Machine Translation, Conference Proceedings* Budapest, Hungary, pp.119–125.

Iyer, R. and Ostendorf, M. 1996. Modelling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *IEEE Transactions on Speech and Audio Processing* pp.236–239.

Kneser, R. and Ney, H. 1995. Improved Backing-off for M-gram Language Modeling In *IEEE International Conference on Acoustics, Speech, and Signal Processing. vol.1* pp.181–184.

Kneser, R. and Steinbiss, V. 1993. On the dynamic adaptation of stochastic language models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993. (ICASSP-93.), vol.2* Minneapolis, MN, pp.586–589.

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)* Barcelona, Spain, pp.388–395.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X: The Tenth Machine Translation Summit* Phuket, Thailand, pp.79–86.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin A. and Herbst H. 2007. Moses: open source toolkit for statistical machine translation. In *ACL-2007: Proceedings of demo and poster sessions* Prague, Czech Republic, pp.177–180.

Koehn, P. and Schroeder, J. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation* Prague, Czech Republic, pp.224–227.

Kuhn, R. and De Mori, R. 1990. A cache-based natural language model for speech recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol.12, no.6, pp.570–583.

Langlais, P. 2002. Improving a general-purpose statistical translation engine by terminological lexicons. In *Proceedings of Coling-2002: Second international workshop on computational terminology (COMPUTERM 2002)* Taipei, Taiwan, pp.1–7.

Och, F. J. and H. Ney 2003. A Systematic Comparison of Various Statistical Alignment Models In *Computational Linguistics* volume 29, (1), pp. 19–51.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association For Computational Linguistics - Volume 1* Sapporo, Japan, pp.160–167.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. 2002. BLEU: a method for automatic evaluation of machine translation In *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics* Philadelphia, PA, pp. 311–318

Stroppa, N. and Way., A. 2006. MaTrEx: DCU machine translation system for IWSLT 2006. In *Proceedings of the International Workshop on Spoken Language Translation* Kyoto, Japan, pp. 31–36.

Wu, H., Wang, H., and Zong, C. (2008). Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Coling 2008, 22nd International Conference on Computational Linguistics* Manchester, UK. pp 993–1000