Experiments in Domain Adaptation for Statistical Machine Translation

Philipp Koehn and Josh Schroeder
pkoehn@inf.ed.ac.uk, j.schroeder@ed.ac.uk
School of Informatics
University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW
Scotland, United Kingdom

Abstract

The special challenge of the WMT 2007 shared task was domain adaptation. We took this opportunity to experiment with various ways of adapting a statistical machine translation systems to a special domain (here: news commentary), when most of the training data is from a different domain (here: European Parliament speeches). This paper also gives a description of the submission of the University of Edinburgh to the shared task.

1 Our framework: the Moses MT system

The open source Moses (Koehn et al., 2007) MT system was originally developed at the University of Edinburgh and received a major boost through a 2007 Johns Hopkins workshop. It is now used at several academic institutions as the basic infrastructure for statistical machine translation research.

The Moses system is an implementation of the phrase-based machine translation approach (Koehn et al., 2003). In this approach, an input sentence is first split into text chunks (so-called phrases), which are then mapped one-to-one to target phrases using a large phrase translation table. Phrases may be reordered, but typically a reordering limit (in our experiments a maximum movement over 6 words) is used. See Figure 1 for an illustration.

Phrase translation probabilities, reordering probabilities and language model probabilities are combined to give each possible sentence translation a score. The best-scoring translation is searched for by the decoding algorithm and outputted by the system as the best translation. The different system components h_i (phrase translation probabilities, language model, etc.) are combined in a log-linear model to obtain the score for the translation **e** for an input sentence **f**:

$$score(\mathbf{e}, \mathbf{f}) = \exp \sum_i \lambda_i \; h_i(\mathbf{e}, \mathbf{f}) \qquad (1)$$

The weights of the components λ_i are set by a discriminative training method on held-out development data (Och, 2003). The basic components used in our experiments are: (a) two phrase translation probabilities (both $p(e|f)$ and $p(f|e)$), (b) two word translation probabilities (both $p(e|f)$ and $p(f|e)$), (c) phrase count, (d) output word count, (e) language model, (f) distance-based reordering model, and (g) lexicalized reordering model.

For a more detailed description of this model, please refer to (Koehn et al., 2005).

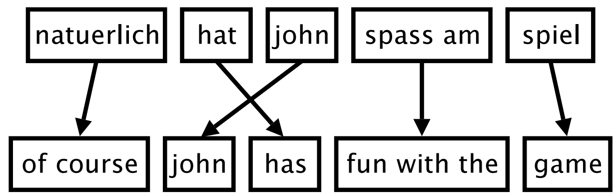

Figure 1: Phrase-based statistical machine translation model: Input is split into text chunks (phrases) which are mapped using a large phrase translation table. Phrases are mapped one-to-one, and may be reordered.

2 Domain adaption

Since training data for statistical machine translation is typically collected opportunistically from wherever it is available, the application domain for a machine translation system may be very different from the domain of the system's training data.

For the WMT 2007 shared task, the challenge was to use a large amount of out-of-domain training data

Proceedings of the Second Workshop on Statistical Machine Translation, pages 224–227,
Prague, June 2007.

(about 40 million words) combined with a much smaller amount of in-domain training data (about 1 million words) to optimize translation performance on that particular domain. We carried out these experiments on French–English.

2.1 Only out-of-domain training data

The first baseline system is trained only on the out-of-domain Europarl corpus, which has the following corpus statistics:

	French	English
Sentences	1,257,419	
Words	37,489,556	33,787,890

2.2 Only in-domain training data

The second baseline system is trained only on the in-domain NewsCommentary corpus. This corpus is much smaller:

	French	English
Sentences	42,884	
Words	1,198,041	1,018,503

2.3 Combined training data

To make use of all the training data, the straight-forward way is to simply concatenate the two training corpora and use the combined data for both translation model and language model training. In our situation, however, the out-of-domain training data overwhelms the in-domain training data due to the sheer relative size. Hence, we do not expect the best performance from this simplistic approach.

2.4 In-domain language model

One way to force a drift to the jargon of the target domain is the use of the language model. In our next setup, we used only in-domain data for training the language model. This enables the system to use all the translation knowledge from the combined corpus, but it gives a preference to word choices that are dominant in the in-domain training data.

2.5 Interpolated language model

Essentially, the goal of our subsequent approaches is to make use of all the training data, but to include a preference for the in-domain jargon by giving more weight to the in-domain training data. This and the next approach explore methods to bias the language model, while the final approach biases the translation model.

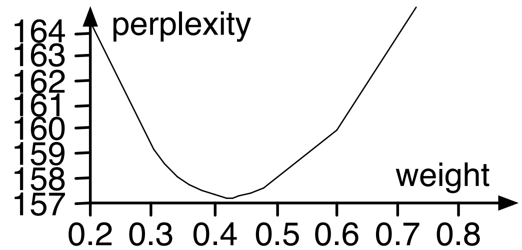

Figure 2: Interpolating in-domain and out-of-domain language models: effect of interpolation weight on perplexity of LM on development set.

We trained two language models, one for each the out-of-domain and the in-domain training data. Language modeling software such as the SRILM toolkit we used (Stolke, 2002) allows the interpolation of these language models. When interpolating, we give the out-of-domain language model a weight in respect to the in-domain language model.

Since we want to obtain a language model that gives us the best performance on the target domain, we set this weight so that the perplexity of the development set from that target domain is optimized. We searched for the optimal weight setting by simply testing a set of weights and focusing on the most promising range of weights.

Figure 2 displays all the weights we explored during this process and the corresponding perplexity of the resulting language model on the development set (nc-dev2007). The optimal weight can be picked out easily from this very smooth curve.

2.6 Two language models

The log-linear modeling approach of statistical machine translation enables a straight-forward combination of the in-domain and out-of-domain language models. We included them as two separate features, whose weights are set with minimum error rate training. The relative weight for each model is set directly by optimizing translation performance.

2.7 Two translation models

Finally, besides biasing the language model to a specific target domain, we may also bias the translation model. Here, we take advantage of a feature of the Moses decoder's factored translation model framework. In factored translation models, the representa-

Method	%BLEU
Large out-of-domain training data	25.11
Small in-domain training data	25.88
Combined training data	26.69
In-domain language model	27.46
Interpolated language model	27.12
Two language models	27.30
Two translation models	27.64

Table 1: Results of domain adaptation experiments

tion of words is extended to a vector of factors (e.g., surface form, lemma, POS, morphology).

The mapping of an input phrase to an output phrase is decomposed into several translation and generation steps, each using a different translation or generation table, respectively. Such a decomposition is called a decoding path.

A more recent feature of the factored translation model framework is the possible use of multiple alternative decoding paths. This alternate decoding path model was developed by Birch et al. (2007). For our purposes, we use two decoding paths, each consisting of only one translation step. One decoding path is the in-domain translation table, and the other decoding path is the out-of-domain translation table. Again, respective weights are set with minimum error rate training.

3 Domain adaptation results

Table 1 shows results of our domain adaptation experiments on the development test set (nc-devtest-2007). The results suggest that the language model is a useful tool for domain adaptation. While training on all the data is essential for good performance, using an in-domain language model alone already gives fairly high performance (27.46). The performance with the interpolated language model (27.12) and two language models (27.30) are similar. All perform better than the three baseline approaches.

The results also suggest that higher performance can be obtained by using two translation models through the Moses decoder's alternative decoding path framework. We saw our best results under this condition (27.64).

4 WMT 2007 shared task submissions

We participated in all categories. Given the four language pairs, with two translation directions and (except for Czech) two test domains, this required us to build 14 translation systems.

We had access to a fairly large computer cluster to carry out our experiments over the course of a few weeks. However, speed issues with the decoder and load issues on the crowded cluster caused us to take a few shortcuts. Also, a bug crept in to our English–French experiments where we used the wrong detokenizer, resulting drop of 2–3 points in %BLEU.

4.1 Tuning

Minimum error rate training is the most time-consuming aspects of the training process. Due to time constraints, we did not carry out this step for all but the Czech systems (a new language for us). For the other systems, we re-used weight settings from our last year's submission.

One of the most crucial outcomes of tuning is a proper weight setting for output length, which is especially important for the BLEU score. Since the training corpus and tokenization changed, our re-used weights are not always optimal in this respect. But only in one case we felt compelled to manually adjust the weight for the word count feature, since the original setup led to a output/reference length ratio of 0.88 on the development test set.

4.2 Domain adaptation

For the Europarl test sets, we did not use any domain adaptation techniques, but simply used either just the Europarl training data or the combined data — whatever gave the higher score on the development test set, although scores differed by only about 0.1–0.2 %BLEU.

In order to be able to re-use the old weights, we were limited to domain adaptation methods that did not change the number of components. We decided to use the interpolated language model method described in Section 2.5. For the different language pairs, optimal interpolation weights differed:

Language pair	Weight for Europarl LM
French–English	0.43
Spanish–English	0.41
German–English	0.40
English–French	0.51
English–Spanish	0.42
English–German	0.45

Language pair	Europarl			NewsCommentary		
	%BLEU	Length	NIST	%BLEU	Length	NIST
French–English	32.66	0.96	7.94	28.27	1.03	7.50
Spanish–English	33.26	1.00	7.82	34.17	1.06	8.35
German–English	28.49	0.94	7.32	25.45	1.01	7.19
Czech–English	–	–	–	22.68	0.98	6.96
English–French	26.76	1.08	6.66	24.38	1.02	6.73
English–Spanish	32.55	0.98	7.66	33.59	0.94	8.46
English–German	20.59	0.97	6.18	17.06	1.00	6.04
English–Czech	–	–	–	12.34	1.02	4.85

Table 2: Test set performance of our systems: BLEU and NIST scores, and output/reference length ratio.

4.3 Training and decoding parameters

We tried to improve performance by increasing some of the limits imposed on the training and decoding setup. During training, long sentences are removed from the training data to speed up the GIZA++ word alignment process. Traditionally, we worked with a sentence length limit of 40. We found that increasing this limit to about 80 gave better results without causing undue problems with running the word alignment (GIZA++ increasingly fails and runs much slower with long sentences).

We also tried to increase beam sizes and the limit on the number of translation options per coverage span (ttable-limit). This has shown to be successful in our experiments with Arabic–English and Chinese–English systems. Surprisingly, increasing the maximum stack size to 1000 (from 200) and ttable-limit to 100 (from 20) has barely any effect on translation performance. The %BLEU score changed only by less than 0.05, and often worsened.

4.4 German–English system

The German–English language pair is especially challenging due to the large differences in word order. Collins et al. (2005) suggest a method to reorder the German input before translating using a set of manually crafted rules. In our German–English submissions, this is done both to the training data and the input to the machine translation system.

5 Conclusions

Our submission to the WMT 2007 shared task is a fairly straight-forward use of the Moses MT system using default parameters. In a sense, we submitted a baseline performance of this system. BLEU and NIST scores for all our systems on the test sets are displayed in Table 2. Compared to other submitted systems, these are very good scores, often the best or second highest scores for these tasks.

We made a special effort in two areas: We explored domain adaptation methods for the NewsCommentary test sets and we used reordering rules for the German–English language pair.

Acknowledgments

This work was supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022 and in part under the EuroMatrix project funded by the European Commission (6th Framework Programme).

References

Birch, A., Osborne, M., and Koehn, P. (2007). CCG supertags in factored statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, Prague. Association for Computational Linguistics.

Collins, M., Koehn, P., and Kucerova, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. of the International Workshop on Spoken Language Translation*.

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, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In Hinrichs, E. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Stolke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.