SEVENTH FRAMEWORK PROGRAMME

THEME ICT-2011.4.2(a)

Language Technologies

ACCEPT

Automated Community Content Editing PorTal

www.accept-project.eu

Starting date of the project: 1 January 2012

Overall duration of the project: 36 months

Report on robust machine translation: domain adaptation and linguistic back-off

Workpackage n° 4	Name: Improving SMT
Deliverable n° 4.2	Name: Report on robust machine translation: domain adaptation and linguistic back-off
Due date: 31 December 2013	Submission date: 19 December 2013

Dissemination level: PU

Organisation name of lead contractor for this deliverable: University of Edinburgh

The research leading to these results has received funding from the European Community's Seventh Framework Programme (*FP7/2007-2013*) under *grant agreement n° 288769.*

Contents

Robust Machine Translation - Domain Adaptation and Linguistic Backoff

1 Overview

This document summarises the research done within the ACCEPT project in *Task 4.2: Domain Adaptation* and *Task 4.3: Linguistic Backoff* of the *Improving SMT* work package. The main body of the research is described in the research papers (fully or partially supported by ACCEPT) available in the appendix. The purpose of this report is to explain how all the research links together, and offer some further experiments on ACCEPT proprietary data.

In SMT, the problem of *domain adaptation* occurs when there is a systematic difference between training and test data, because they come from different sources. For example, the training data might be drawn mainly from parliamentary proceedings but we want to use the resulting system to translate public health information leaflets. Normally training data which is similar to the test data is know as *in-domain* and training data different from the test data is known as *out-of-domain*.

The idea of linguistic backoff is to solve a specific problem with translating from morphologically rich languages. The problem is that the morphological complexity causes data sparsity since many possible surface forms are not observed in training. In linguistic backoff we use factored models and linguistic analysis to synthesise translations for these unobserved forms.

2 Domain Adaptation

2.1 Methods

An overview of methods for domain adaptation is shown in Figure 1. This is not necessarily comprehensive, but shows the methods considered in this report. In fact we mainly focus on methods which improve the scoring or the model, and methods which improve the training data for the translation system. Methods to obtain more data are also important but we do not consider them here.

The first piece of work on domain adaptation that we published in the ACCEPT project (Haddow and Koehn 2012) was an analysis of how domain change affects performance in SMT. Through experiments on two domain adaptation problems in 8 language pairs, we showed that the biggest contribution of in-domain data was reducing the number of out-of-vocabulary words, but that improving the scoring of seen words was also important. Furthermore, we showed that adding out-of-domain data generally helped with unseen or rarely seen words, but could degrade performance on moderately frequent words.

2.2 Instance Selection

The idea of *instance selection* (also know as *subsampling* or *data filtering*) is that to build a domain-specific translation system, we should be selective about which parallel sentences to include in the training data. In other words, we should filter out sentences which could degrade translation quality by causing the model to prefer incorrect translations. Filtering

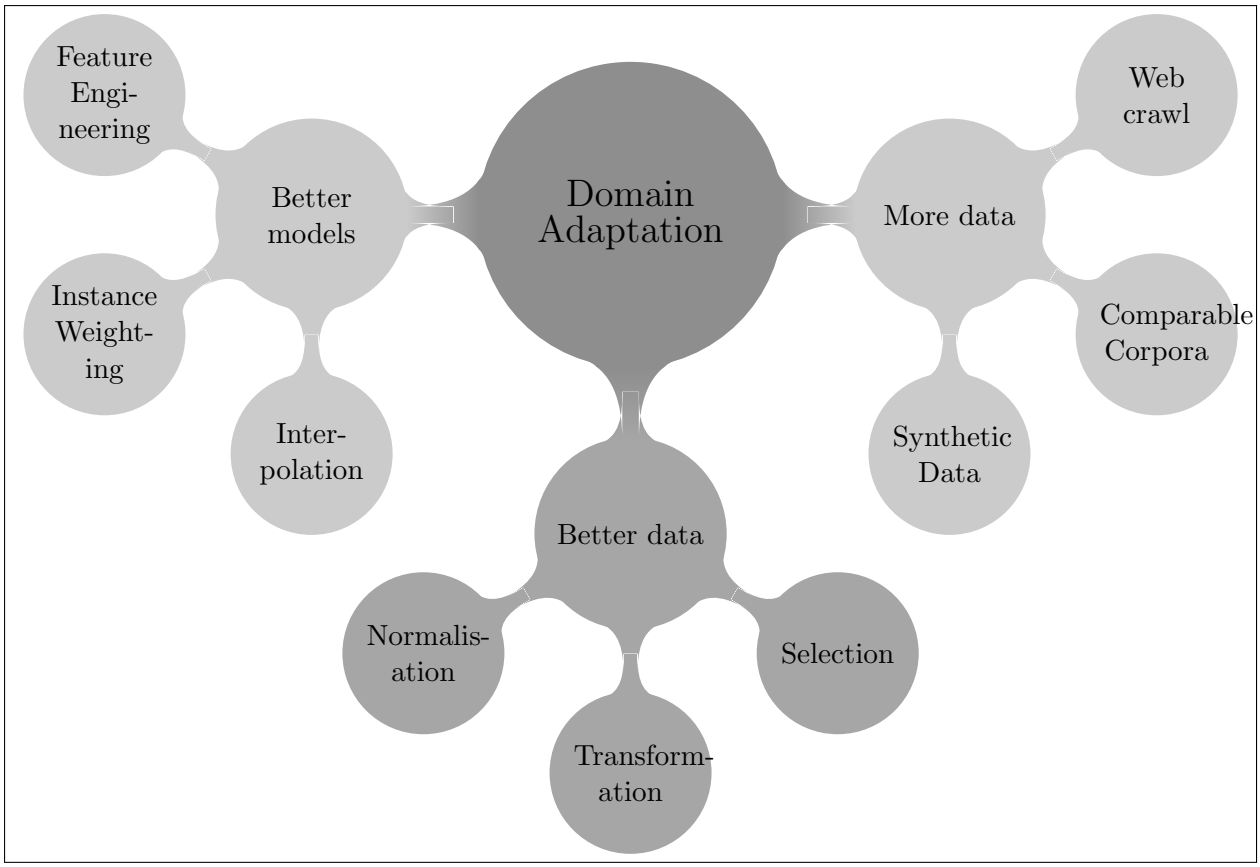


Figure 1: Domain adaptation methods

methods from the literature typically try to find parallel training sentences which are in some way similar to the type of data we would like to translate, using n-gram overlap to define similarity.

One filtering method which has been successful is known as *Modified Moore-Lewis* (Moore and Lewis 2010; Axelrod et al. 2011). The idea here is to select data from a collection of "general domain" parallel text, picking those sentence pairs which are similar to a collection of in-domain data, but different from the general domain text. Similarity is measured by language model perplexity, which roughly corresponds to n-gram overlap.

We implemented MML filtering in Moses, and tested in on the WMT (Workshop in Machine Translation) shared tasks in 2012[1] and 2013[2]. For the 2012 shared task (Koehn and Haddow 2012b), we found that MML filtering was somewhat effective in the Spanish↔English translation task, but not in the French↔English task. It was shown to be more effective than filtering using IBM Model 1 score and (not shown in the paper) in-domain language model perplexity, and random filtering.

For WMT2013, we tested MML filtering on all the WMT language pairs apart from German↔English, and results are available in Durrani et al. (2013) (see Table 14). In this extended test, it was again found to be really only effective on the Spanish↔English tasks. Experiments with MML on ACCEPT data are shown in Section 2.7.2

In Banerjee et al. (2013) we developed an improvement on the MML approach. In this work, instead of designating a given corpus as "out-of-domain", we use quality estimation to determine where the system performs badly, and regard this as the out-of-domain

[1] `www.statmt.org/wmt12`
[2] `www.statmt.org/wmt13`

data set. In experiments on the ACCEPT Symantec data, best results (in BLEU) are comparable to standard MML, but are achieved with less data.

2.3 Instance Weighting

If instance selection could be considered as a applying a 1-0 weighting to the parallel sentences in the pool of training data, then its natural generalisation is *instance weighting*, where arbitrary weights are allowed. We have implemented a mechanism in the Moses training pipeline to allow weighting of training sentences in the translation and reordering model, and experimented with weights derived from the MML scores. These experiments were also within the context of the WMT2013 shared task, and the results are shown in Table 14 of our system paper (Durrani et al. 2013). The results show that instance weighting overall performs a bit better than instance selection, but the results are again quite variable.

A better approach than this heuristic weighting of the sentences would be to train these weights somehow, ideally to maximise translation performance. In Section 2.4, a method to train mixture model weights to maximise performance (PRO-MIX) is described and we propose to extend this to train instance weights. This has been partly implemented, but there is still a significant amount of work required in order to obtain results from this approach.

2.4 Mixture Models

In this approach to domain adaptation, we build separate models on portions of the data, and interpolate them to produce a single model, selecting the interpolation weights to optimise performance on the desired domain.

For language modelling, this is our usual approach. We build separate language models on each of the corpora that we are using to build the system, and then interpolate them linearly to minimise perplexity on a heldout set of in-domain data (usually the tuning set). This method of weight estimation is implemented in the popular SRILM toolkit. This has been shown to be effective in earlier work, e.g. (Koehn and Schroeder 2007), so in the experiments in this report we always use language models created in this manner.

The interpolation of translation models, however, has received less attention (Foster and Kuhn 2007; Banerjee et al. 2011; Sennrich 2012). Again, it is possible to optimise the interpolation weights for perplexity on a heldout set, and implementations from Sennrich (2012) are available in Moses. For the WMT13 shared task, we ran experiments with both the "naive" and "modified" versions of this implementation of interpolation. The naive method simply interpolates the 4 phrase features, whereas the modified method handles OOVs and the lexical weights in a more principled fashion. The WMT13 experiments (shown in Table 13 of Durrani et al. (2013)) showed that interpolation increased BLEU more often than it decreased it, but there was no clear difference between naive and modified interpolation. Experiments using these interpolation methods with ACCEPT data are shown in Section 2.7.4.

Optimising interpolation weights for perplexity is relatively straightforward to implement, but it is unclear how well translation model perplexity correlates with translation performance. Translation model perplexity also is not well-defined when the two models have different supports (i.e. different coverages), which they invariably do. For these reasons we decided to investigate whether translation model interpolation weights could be

optimised directly for a measure of translation performance, such as BLEU. This was found to be possible using a modification of the Pairwise Ranked Optimisation (PRO) training algorithm, and experiments across two data sets and several language pairs showed it to be better than perplexity minimisation (Haddow 2013). We have yet to try this method on the WMT or ACCEPT data, as the current implementation of the PRO-MIX training only enables the interpolation of 2 translation models, although the method itself is not limited in this way.

As mentioned in the previous section, we propose to extend the PRO-MIX method to train weights for all sentences of the corpus. The idea is that a sentence's provenance (i.e. which corpus it is found in) is only one piece of information, and it should really be used as a feature the model, along with other aspects of the sentence (for example, topic, alignment quality, immediate context, date etc.).

2.5 Data Transformation

Instead of obtaining more in-domain data to build our translation system with, in some cases it may be possible to transform the out-of-domain training data to look more like the in-domain test data. Or vice-versa, i.e. we could transform in-domain test data to look more like the out-of-domain training data.

This was the focus of Rayner et al. (2012), in which we considered the issue of register mismatch between training and test data. Specifically, we noted that when trying to translate text from the Symantec forums in French, the europarl training data often had poor coverage of informal (i.e. second person singular) verb-forms and also informal pronouns. In the Rayner et al. (2012) paper we developed transformation rules to convert French informal verb forms into their formal equivalents (and vice-versa) and to transform French questions between formal and the informal "est-ce que" form. The former transformation was found to be helpful, whether we ran it on the test data or (in reverse) on the training data, whereas transformation of questions turned out to be more complex and not amenable to this approach. Later experiments showed that transforming *both* training and test data was significantly better than translating just one of them. We have also experimented with using the linguistic backoff approach of Section 3 to address the lack of coverage of French informal verb-forms, but found that it did not give good results because it does not take into account the source context.

In further work on the transformation of informal text (Bouillon et al. 2013) we looked at ways of correcting spelling errors in French. Specifically, we focused on spelling errors caused by homophone confusions and compared an approach using manually written pre-editing rules, with confusion networks constructed using a pronunciation dictionary. Both methods were judged to be effective, and somewhat complementary, however further analysis of the results has revealed issues with the quality of the data (Symantec French-English) and we are currently improving this with a view to writing an extended version of Bouillon et al. (2013) for journal publication.

2.6 Feature Engineering

The final technique that we investigated was enabled by the availability of new discriminative training algorithms(Hopkins and May 2011; Cherry and Foster 2012), with implementations in Moses. These new algorithms allow us to train SMT models with much more features than the 15-20 allowed by MERT, and to investigate the usage of such

features for domain adaptation.

A relatively simple example of feature engineering for domain adaptation is to attach *indicator features* to each phrase in the phrase table to show which corpus (or corpora) it was found in. We applied this technique in the WMT 2013 shared task (see Durrani et al. (2013) - Table 4) with some success. We have also tested it on ACCEPT data, with results shown in Section 2.7.5.

Experiments with a much larger feature set for domain adaptation were performed in Hasler et al. (2012), on data for the IWSLT shared task. In this work we used both sparse word-pair features, and topic-word-pair features, again with small positive results. The work on using topic models for domain adaptation is ongoing, with partial support from the ACCEPT project.

2.7 ACCEPT Experiments

In this section we report on the performance of various domain adaptation techniques on data from the ACCEPT projects. We consider two different target domains (Translators Without Borders and Symantec), where the test data consists of health-care documents translated by TWB volunteers, and Symantec user forum data, respectively. The training data consists of data released for the WMT shared task, as well as data provided by the relevant ACCEPT partners.

To create the TWB test data, we took parallel pairs of French-English documents translated by the TWB volunteers (which were initially in various Microsoft formats), extracted and sentence-aligned the text, and randomly selected documents that were manually judged to be on the health-care topic. The sentence-alignment of the test set was checked and corrected by French-English translators at the University of Geneva. We extracted two test sets, one where the documents were originally in English, and one where they were originally in French.

The Symantec test sets were created from posts collected from the Symantec user forums. The posts were pre-processed by Symantec to replace some URLs and PATHs with placeholders. The English→German and English→French sets were created by randomly selecting a set of English posts, and translating them into German and French. Some translations were done by using Google translate and post-editing, and some were translated from scratch, but all translation was done by professional translators familiar with Symantec content. The French→English set was created by the University of Geneva, by post-editing translations from Google translate.

Statistics on the test sets are shown in Table 1. Note that each of the test sets was split randomly into equal sized tuning and test sets, preserving document boundaries where necessary, so the experimental results are reported on data sets half the size of those in the table.

The training data for the experiments consists of the following corpora:

news-commentary News analysis data from the WMT2012 shared task.

europarl European parliamentary proceedings from the WMT2012 shared task.

emea Data from the European Medicines Agency, released by the OPUS[3] project.

symantec-tm Translation memories from Symantec, containing product manuals and marketing materials. This is not strictly in-domain, especially as its register is quite different from the forum data.

symantec-forum A monolingual collection of posts from the Symantec forums.

[3] `http://opus.lingfil.uu.se`

Domain	Pair	Sentences	Words	
			Source	Target
Symantec	en-fr	3031	49715	54217
	en-de	3031	49715	46184
	fr-en	1022	20435	18173
TWB	fr-en	2004	32983	31180
	en-fr	2253	37436	43142

Table 1: Combined tuning and test set statistics for (un-preprocessed) ACCEPT data.

TWB Documents translated by TWB volunteers. The TWB test sets are drawn from this corpus, with the remainder used for training.

The statistics of these data sets are shown in Table 2.

Data Set	Language	Sentences	Words	
			Source	Target
news-commentary	fr-en	137097	3449140	2991501
	de-en	158840	3492906	3390291
europarl	fr-en	2007723	52525000	50263003
	de-en	1920209	44613020	47881421
emea	fr-en	1092568	14144280	12195082
symantec-tm	fr-en	1614081	17203958	14755674
	de-en	1904301	17945253	18378277
TWB	fr-en	316388	4141138	3793871
symantec-forum	en	353047	4983671	
	de	106028	1303497	
	fr	107407	1344923	

Table 2: Training data sets (monolingual and bilingual) used for the experiments on ACCEPT data

2.7.1 Baselines

In all the experiments we used a standard Moses phrase-based system, with default settings. The tuning was performed with batch MIRA (Cherry and Foster 2012), and performance was measured with case-sensitive BLEU using the `multi-bleu.perl` script provided with Moses.

The Symantec systems were built using the news-commentary, europarl, symantec-forum and symantec-tm corpora, whereas the TWB systems were built with the news-commentary, europarl, emea and TWB data sets. For the baselines, the parallel data sets were concatenated together to train the translation models. For the language models, separate models were created from each corpora, which were then interpolated, minimising perplexity on the tuning set.

The first thing that was noted in the baseline systems, mainly for the Symantec systems, is that there were a lot of URLs and directory names, which were being split up by the tokeniser and translated separately. This meant that we were observing a lot of uninteresting errors on URL fragments, and also that BLEU was higher than it should

be since we got credit for translating each piece of the URL correctly. To prevent this from happening, we modified the Moses tokeniser to force it to ignore certain patterns representing URLs, pathnames and Windows registry keys.

Another observation on the initial baseline run was that there were frequent case errors in the Symantec data. To reduce these, we added the English symantec-forum data to the training data for the Moses truecaser, and this improved BLEU scores by 0.5-0.76. The definitive baseline scores are shown in Table 3. (Note that all BLEU scores quoted are the result of averaging over 3 runs of tuning)

	Symantec			TWB	
	en-fr	en-de	fr-en	fr-en	en-fr
Baseline	32.05	19.51	42.79	34.66	26.60

Table 3: Baseline BLEU scores

2.7.2 Instance Selection

We report on experiments using MML (Section 2.2) to select parallel data for training the translation model.

For Symantec we treated all except for the symantec-tm as out-of-domain, and similarly for TWB we treated all except TWB as out-of-domain. For Symantec we used the forum data to build the in-domain language model, whereas for TWB we just used both sides of the parallel TWB data. We experimented with selecting 10%, 20%, 50% and 80% of all available out-of-domain data, and results are shown in Table 4

	Symantec			TWB	
% retained	en-fr	en-de	fr-en	fr-en	en-fr
10%	31.96 (-.08)	19.30 (-.21)	42.13 (-.66)	34.69 (+.03)	26.76 (+.16)
20%	32.35 (+.30)	19.26 (-.25)	42.61 (-.18)	34.73 (+.07)	26.86 (+.26)
50%	32.21 (+.16)	19.20 (-.31)	42.64 (-.15)	34.81 (+.15)	26.92 (+.32)
80%	32.10 (+.05)	19.18 (-.33)	42.77 (-.02)	34.83 (+.17)	26.66 (+.06)

Table 4: MML selection scores for ACCEPT data, and different selection ratios. Figures in brackets show change relative to baseline

2.7.3 Instance Weighting

For the instance weighting experiments, we used weights derived from the MML score. We first subtract the maximum score from each MML score, divide by a scale factor and then apply the exponential function. In the Symantec case the weighting was applied to all parallel data, since none of it was really "in-domain". However for the TWB system, we did not scale the TWB data, but gave it all a weight of 1 (the maximum)

We tried two different scale factors, and the results are available in Table 5.

	Symantec			TWB	
scale	en-fr	en-de	fr-en	fr-en	en-fr
2	31.97 (-.08)	19.27 (-.24)	42.81 (+.03)	35.15 (+.49)	26.41 (-.19)
10	32.13 (+.08)	19.53 (+.02)	43.23 (+.44)	35.14 (+.48)	26.91 (+.31)

Table 5: Instance weighting scores for ACCEPT data, using MML derived weights, with different weightings. Figures in brackets show change relative to baseline.

2.7.4 Mixture Models

We compared two methods of interpolating translation models, the *naive* and *modified* methods of Sennrich (2012). These both use perplexity minimisation to train the interpolation weights. We used the alignments derived from the concatenation of all the corpora, then created separate translation models for each corpus and then interpolated them. The results are shown in Table 6.

	Symantec			TWB	
method	en-fr	en-de	fr-en	fr-en	en-fr
naive	32.13 (+.08)	19.44 (-.07)	43.31 (+.52)	34.89 (+.23)	27.05 (+.45)
modified	32.02 (-.03)	19.59 (+.08)	43.08 (+.29)	35.16 (+.50)	26.73 (+.13)

Table 6: Linear interpolation scores for ACCEPT data, using different types of perplexity minimisation. Figures in brackets show change relative to baseline.

2.7.5 Domain Indicator Features

Finally we show the effect of using domain indicator features. In this case, the system is the same as the baseline, except that each phrase-pair has an extra set of features associated with it indicating which corpus or corpora it was found in. The results are shown in Table 7.

Symantec			TWB	
en-fr	en-de	fr-en	fr-en	en-fr
32.32 (+.28)	19.30 (-.21)	43.23 (+.44)	34.94 (+.28)	27.16 (+.56)

Table 7: Domain indicator features for ACCEPT data. Figures in brackets show change relative to baseline.

2.7.6 Analysis

None of the techniques emerged as a clearcut winner in the experiments above. The instance selection shows improvement on 3 of the 5 data sets, although the absolute improvement is small (about +0.3 BLEU) and careful choice of the filtering percentage is required. For the other techniques, which all attempt to rescore the phrase table in some way, the effects are mostly positive or neutral, although there are exceptions. The instance weighting (scale 10) and domain indicator show the best average improvement (+0.27 BLEU), and the former is the only technique that never has a negative effect.

In order to gain some insight beyond the BLEU score, we applied the S^4 scheme for lexical errors proposed in Irvine et al. (2013), and using the WADE analysis code released with that paper. This paper proposes a categorisation of errors into 4 types – although for our purposes we collapse two of the types as they are difficult to separate:

Seen The source word has not been seen in training, i.e. it is out-of-vocabulary (OOV).

Sense The source word has been seen in training, but its reference translation is not in the phrase table.

Score (or Search) The source word and reference translation are in the phrase table, but the decoder chose the wrong one. We included Search errors with Score errors.

In Figure 2 we plot Seen errors versus Score errors for all 5 data sets, for the MML filtering experiments. For each data set, the number of Seen errors increases as we reduce the amount of the data set used (more OOVs), but the number of Score errors decreases, indicating better translation probability estimates. Plotting Sense errors versus Score errors showed a similar pattern. The fact that Score errors reduce with filtering suggests that the filtering is somewhat helpful at reducing bad translations, but the corresponding increase in OOVs may result in an overall negative effect on BLEU score. It is hard to discern any difference between the data sets where MML was effective, and those where it was not. The main difference seems to be between the TWB and Symantec data sets, where there is a much larger effect of the filtering on the former. This may be, however, because the in-domain data set is smaller for TWB than for Symantec

For the methods other than instance selection, i.e. those that aim to improve the phrase table scoring, we show the relationship between score errors and BLEU in Figure 3. For these methods, the numbers of Seen and Sense errors are the same for all models, since the phrase pairs in the phrase table are not changed, just their scores. By plotting the Score errors against BLEU, we seek to determine whether the rescoring of the phrase table is simply not effective, or whether it is effective and does not improve BLEU.

Looking at the plots it is immediately apparent that there is little or no correlation between BLEU and Score errors for either the Symantec en-de or en-fr data sets. Both of these share a common source set, and the S^4 analysis is based on counts of source words. The other three data sets show an approximately linear (inverse) relation between Score errors and BLEU scores, as one might expect, with both the fr-en sets showing reductions in Score errors and increases in BLEU for all techniques.

An analysis of the most common words causing errors of the types examined above, showed some limitations of the WADE method of extracted S^4 scores. Specifically, WADE counts *alignment links*, not source tokens, so this can give misleading results when (e.g.) one source token is aligned with several tokens in the reference. It is possible that this does not make any difference at the macro level, but in order to get a better idea of which words are affected most by the rescoring it would be necessary to modify the WADE method.

3 Linguistic Backoff

When translating from an inflected language, we always face the problem of data sparsity because there will be many inflected forms that we have not seen in training data. As

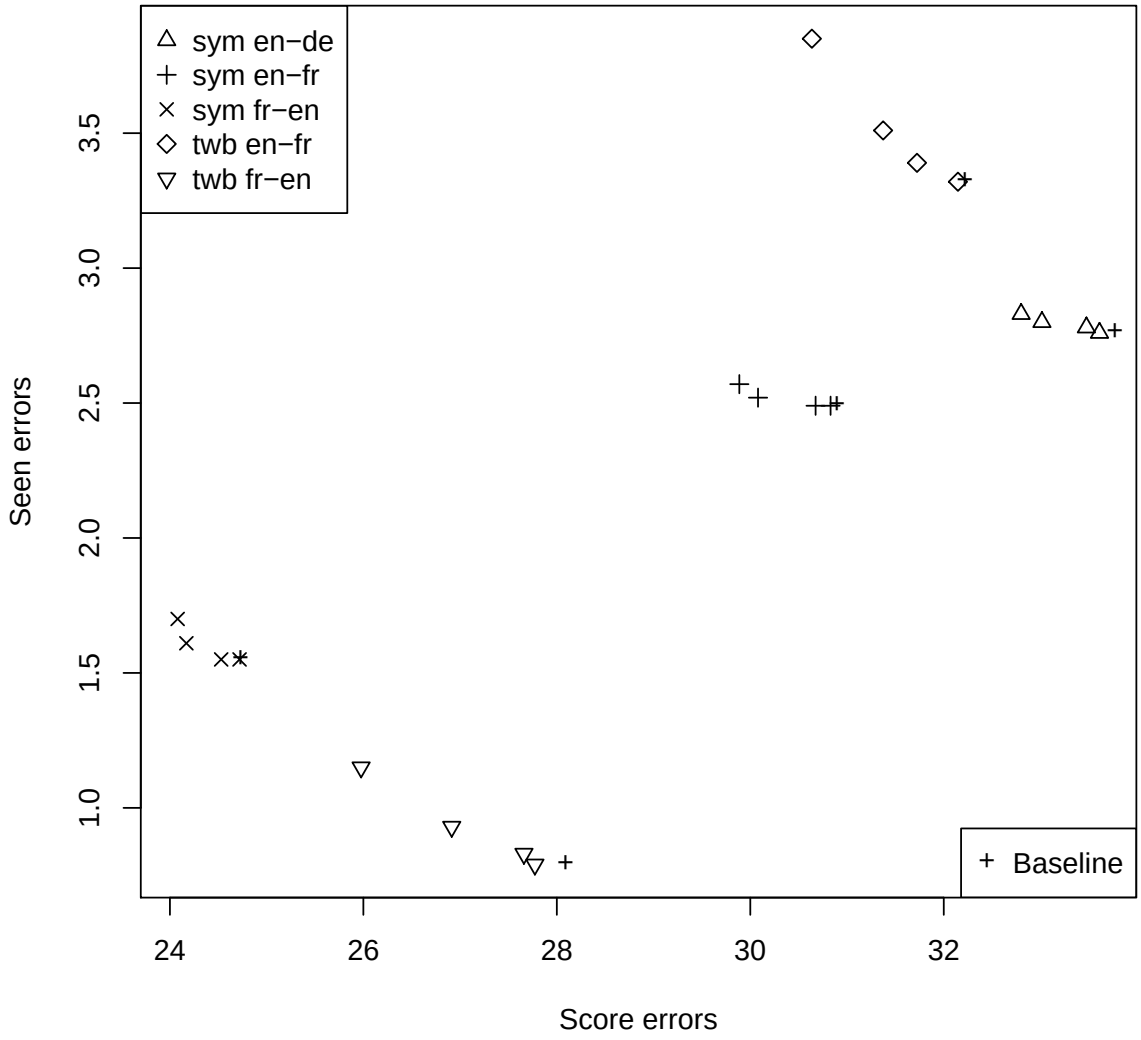


Figure 2: Seen versus Score errors for Modified Moore-Lewis (MML) filtering. In each group, the lower right point is the baseline, and moving up an left shows more aggressive filtering (lower percentage retained).

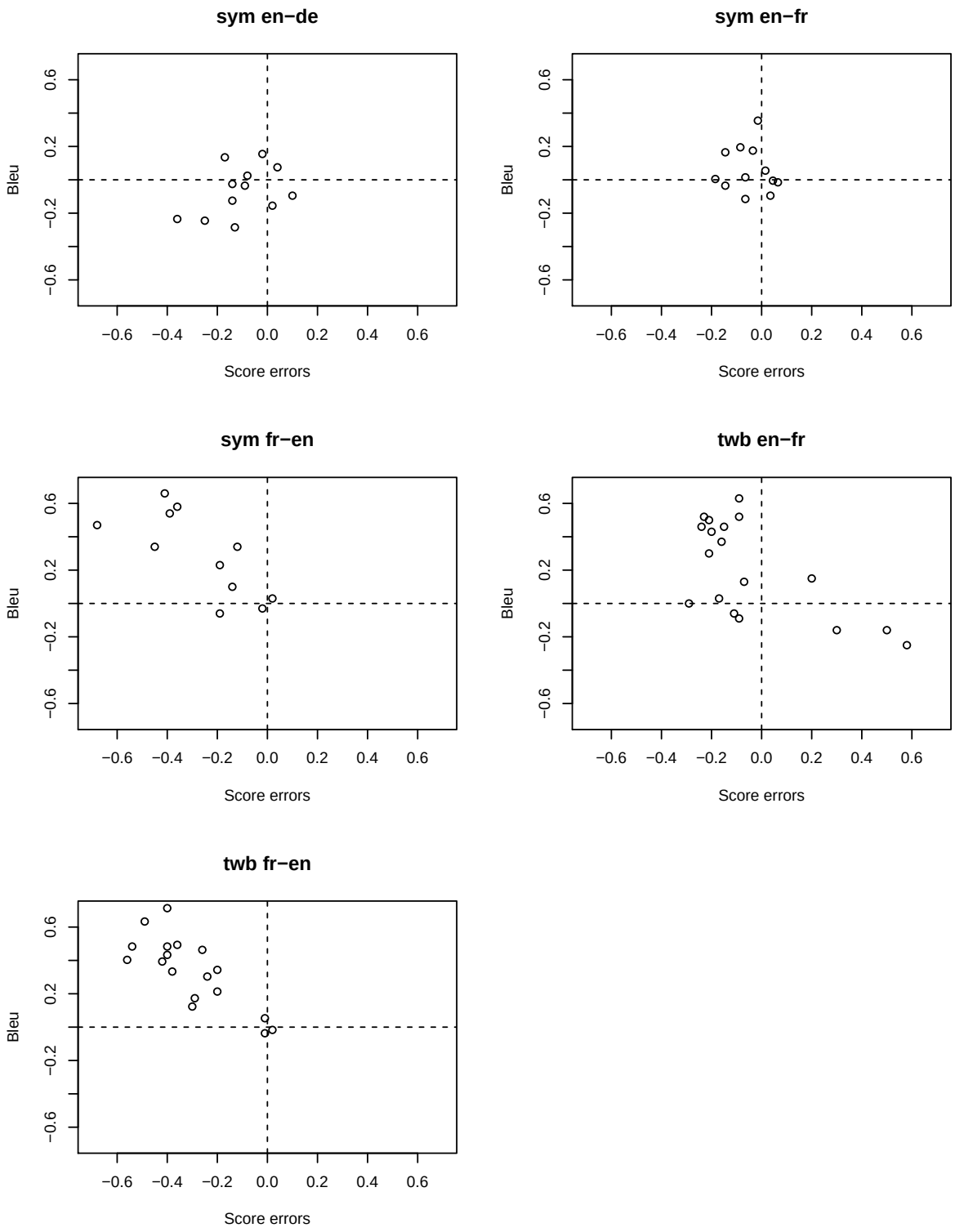


Figure 3: BLEU versus Score errors for all the rescoring domain adaptation methods. All figures are relative to the mean baseline

always, this problem will be exacerbated when the training data set is small. Linguistic backoff is a potential solution to this problem, where if a given surface form is unseen (i.e. OOV) then we attempt to use linguistic analysis to translate it. The method can be extended to help with the translation of rarely seen inflected forms, which may suffer from poor probability estimates due to their rarity.

Linguistic backoff and the extension to interpolated backoff is the topic of Koehn and Haddow (2012a), where we show that it can improve translation from German to English (using WMT data), particularly of forms that are rare or missing in the source side of the training data. The backoff works by using a factored model, where the lemma and morphological tag are translated separately, and then recombined in a generation step to create the final surface form.

4 Conclusions

In this deliverable we presented many methods of domain adaptation and evaluated them on diverse data sets. However there is no one method that consistently delivers improvement, and in fact methods will show positive effects on some data set / language pair combinations but negative effects on other combinations. What this indicates is that the problem of domain adaptation is not fully understood and calls for more work analysing *why* we need domain adaptation (following on from (Haddow and Koehn 2012; Irvine et al. 2013; Banerjee 2013)), and why different techniques work (or not). We also note that many of the techniques in this report address Score errors (in the terminology of the second reference), but in many cases the problem of Seen or Sense errors has a more serious effect of performance. There has been some recent work to address this (e.g. (Banerjee et al. 2012; Irvine et al. 2013)) and it is something that we will continue to investigate in ACCEPT.

References

Axelrod, A., X. He, and J. Gao (2011, July). Domain Adaptation via Pseudo In-Domain Data Selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., pp. 355–362. Association for Computational Linguistics.

Banerjee, P. (2013). *Domain Adapation for Statistical Machine Translation of Corporate and User-Generated Content*. Phd, Dublin City University.

Banerjee, P., S. K. Naskar, J. Roturier, A. Way, J. van Genabith, and J. van Genabith (2011). Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component Level Mixture Modelling. In *Proceedings of MT Summit*.

Banerjee, P., S. K. Naskar, J. Roturier, A. Way, J. van Genabith, and J. van Genabith (2012). Domain Adaptation in SMT of User-Generated Forum Content Guided by OOV Word Reduction: Normalization and/or Supplementary Data? In *Proceedings of EAMT*.

Banerjee, P., R. Rubino, J. Roturier, and J. van Genabith (2013). Quality estimation-guided data selection for domain adaptation of smt. In *Proceedings of MT Summit*.

Bouillon, P., J. Gerlach, U. German, B. Haddow, and M. Rayner (2013). Two Approaches to Correcting Homophone Confusions in a Hybrid Machine Translation

System. In *Proceedings of Second Workshop on Hybrid Approaches to Translation (HyTra)*.

Cherry, C. and G. Foster (2012). Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of NAACL*.

Durrani, N., B. Haddow, K. Heafield, and P. Koehn (2013, August). Edinburgh's machine translation systems for European language pairs. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, pp. 114–121. Association for Computational Linguistics.

Foster, G. and R. Kuhn (2007, June). Mixture-Model Adaptation for {SMT}. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, pp. 128–135. Association for Computational Linguistics.

Haddow, B. (2013). Applying pairwise ranked optimisation to improve the interpolation of translation models. In *Proceedings of NAACL*.

Haddow, B. and P. Koehn (2012, June). Analysing the effect of out-of-domain data on SMT systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Montreal, Canada. Association for Computational Linguistics.

Hasler, E., B. Haddow, and P. Koehn (2012). Sparse lexicalised features and topic adaptation for SMT. In *Proceedings of IWSLT*.

Hopkins, M. and J. May (2011, July). Tuning as Ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., pp. 1352–1362. Association for Computational Linguistics.

Irvine, A., J. Morgan, M. Carpuat, H. D. III, and D. Munteanu (2013). Measuring machine translation errors in new domains. *Transactions of the ACL Q13*, 1035.

Irvine, A., C. Quirk, and H. Daumé III (2013, October). Monolingual marginal matching for translation model adaptation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1077–1088. Association for Computational Linguistics.

Koehn, P. and B. Haddow (2012a). Interpolated backoff for factored translation models. In *Proceedings of AMTA*.

Koehn, P. and B. Haddow (2012b, June). Towards effective use of training data in statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Montreal, Canada. Association for Computational Linguistics.

Koehn, P. and J. Schroeder (2007, June). Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, pp. 224–227. Association for Computational Linguistics.

Moore, R. C. and W. Lewis (2010, July). Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden, pp. 220–224. Association for Computational Linguistics.

Rayner, M., P. Bouillon, and B. Haddow (2012). Using source-language transformations to address register mismatches in SMT. In *Proceedings of AMTA*.

Sennrich, R. (2012). Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of EACL*.

A Published Papers

This is the list of papers, fully or partially supported by ACCEPT.

Reference	Support
(Banerjee et al. 2013)	Partial
(Bouillon et al. 2013)	Full
(Durrani et al. 2013)	Partial
(Haddow 2013)	Full
(Haddow and Koehn 2012)	Full
(Hasler et al. 2012)	Partial
(Koehn and Haddow 2012a)	Full
(Koehn and Haddow 2012b)	Partial
(Rayner et al. 2012)	Full

The papers are included on the following pages.

Quality Estimation-guided Data Selection for Domain Adaptation of SMT

Pratyush Banerjee, Raphael Rubino, Johann Roturier[1], Josef van Genabith
CNGL, School of Computing, Dublin City University, Dublin, Ireland
{pbanerjee,rrubino,josef}@computing.dcu.ie
[1] Symantec Research Labs, Dublin, Ireland
johann_roturier@symantec.com

Abstract

Supplementary data selection is a strongly motivated approach in domain adaptation of statistical machine translation systems. In this paper we report a novel approach of data selection guided by automatic quality estimation. In contrast to the conventional approach of using the entire target-domain data as reference for data selection, we restrict the reference set only to sentences poorly translated by the baseline model. Automatic quality estimation is used to identify such poorly translated sentences in the target domain. Our experiments reveal that this approach provides statistically significant improvements over the unadapted baseline and achieves comparable scores to that of conventional data selection approaches with significantly smaller amounts of selected data.

1 Introduction

The quality of translations generated by a statistical machine translation (SMT) system depends heavily on the *amount* of available parallel training data, as well as on the *domain-specificity* of the training and target datasets (Axelrod et al., 2011). Real-life translation tasks are usually domain-specific in nature and require large volumes of in-domain parallel training data. However, such domain-specific parallel training data is often sparse or completely unavailable. In such scenarios, domain adaptation techniques are necessary to effectively leverage available out-of-domain or related-domain parallel data. Supplementary data selection (Hildebrand et al., 2005; Axelrod et al., 2011) is one such popular technique which uses out-of-domain parallel data to supplement sparse in-domain data. However, combining lots of out-of-domain data with small amounts of in-domain data might negatively affect translation quality by overwhelming the in-domain characteristics. Hence *relevant data selection* is used, where only a sub-part of the out-of-domain data, *relevant* to the target domain, supplements the sparse in-domain training data.

Conventionally, the data selection process is guided by all available monolingual (or bilingual) target-domain data. Sentence pairs from out-of-domain data, which are similar (in terms of a similarity metric) to the sentences in the target-domain, are chosen for adaptation with the objective of improving translation quality of all target domain sentences. However, an unadapted baseline system may already translate some target-domain sentences well, thus limiting their scope of improvement by adaptation. In contrast, the sentences poorly translated by the baseline system might have a higher potential for improvement. Utilising this category of target-domain sentences to guide the data selection process forms the primary motivation of our approach.

In order to identify the poorly translated sentences in the target domain, we utilise quality estimation (QE) techniques which involve the process of estimating how good the translation output is, through characteristic elements extracted from the source and the target texts, and also from the SMT system involved (if accessible). These features are predictive parameters derived from the text and associated with quality scores or labels, such as au-

tomatic or manual evaluation scores. When the QE task consists of predicting labels, such as *good* or *bad*, for a given translation pair, classification and/or regression techniques can be used. The classification approach leads to direct label prediction, whereas the regression approach uses an acceptance threshold set on the predicted scores. In our approach, we experiment with both methods using a manually set threshold on the reference dataset. After predicting the poor translations on the target domain, the corresponding source sentences are used to select relevant supplementary parallel training data. In order to highlight the effectiveness of our approach we compare it with a standard technique of data selection based on the entire target-domain data. The experiments reveal that our approach provides improvements comparable to that of standard data selection techniques but with significantly smaller amounts of selected supplementary data.

In this paper we apply our approach to the task of adapting an SMT system to translate user-generated content in the Symantec web forums. The major challenge in translation of forum content lies in the lack of parallel forum-style training data. Hence, we utilise in-domain parallel training data in the form of Symantec translation memories (TMs) as a part of our baseline training data. Symantec TMs comprise internal documentation on Symantec products and services, while the forums consists of user discussions pertaining to the same. Hence, despite being in the same domain the TM data is clean, professionally edited and generally conforms to controlled language guidelines, whereas the forum data is often noisy, user-generated and has a wider vocabulary and colloquialisms. This difference between the training and target datasets necessitates the use of supplementary data for adaptation, thus making this an appropriate use-case for our approach.

The rest of paper is organised as follows: Section 2 presents related work relevant to our approach. Section 3 details the QE and data selection methods. Section 4 presents the experimental setup and results followed by discussions and conclusions in Section 5 and 6, respectively.

2 Related Work

QE for SMT was first applied at the word-level (Ueffing et al., 2003) and then extended to the sentence-level (Blatz et al., 2003). More recently, several studies have focused on using human scores to evaluate the translation quality in terms of post-editing effort (Callison-Burch et al., 2012) or translation adequacy (Specia et al., 2011). The promising results obtained in QE lead to interesting applications in MT, such as sentence-selection for statistical post-editing (Rubino et al., 2012) or system combination (Okita et al., 2012). In this paper, we apply QE techniques to identify *bad* translations from the target domain to drive domain adaptation by data selection.

In order to select supplementary out-of-domain data relevant to the target domain, a variety of criteria have been explored in the MT literature, ranging from information retrieval techniques (Hildebrand et al., 2005) to perplexity on 'in-domain' datasets (Foster and Kuhn, 2007). Axelrod et al. (2011) presented a technique using the bilingual difference of cross-entropy on 'in-domain' and 'out-of-domain' language models for ranking and selection by thresholding, which outperformed the monolingual perplexity based techniques. More recently, Banerjee et al. (2012) presented a novel translation-quality evaluation (rather than prediction) based data selection technique using an incremental translation model merging approach. While all these approaches select data with respect to the entire available target domain data, our approach uses only a sub-part of the same comprising potentially poorly translated sentences. Hence any of these techniques could effectively be combined with our approach. Here, we use the bilingual cross-entropy difference based approach (Axelrod et al., 2011) in our experimental setup. To the best of our knowledge, the QE-guided data selection approach is novel and is one of the primary contributions of this paper.

3 QE-based Data Selection

This section presents the details of the three individual components involved in our approach.

3.1 Automatic Quality Estimation

To distinguish between the good and the bad translations of the target-domain (English forum data in our context), we experimented with both classification as well as regression-based QE approaches. For both sets of experiments, we extract 17 features similar to the baseline QE setup

suggested by the organisers of the WMT12 shared task (Callison-Burch et al., 2012), which were shown to perform well on a post-editing effort prediction task. In our study, we want to predict the Translation Edit Rate (TER) (Snover et al., 2006) to spot *bad* translations. Given the TER scores for a set of translations, identifying the bad translations requires a threshold value, such that all sentences having TER scores above this threshold would be labelled as *bad* translation. However, a translation with a low TER score may still be considered *bad* since TER does not incorporate the notion of semantic equivalence (Snover et al., 2006).

To set the value of this threshold, we selected two sets of 50 sentences randomly from our QE En–Fr training data such that there was an overlap of 10 sentences in each. These sentences along with their manual translations, baseline SMT generated translations and TER scores were reviewed by two evaluators who are native French speakers. The objective of the manual evaluation was to identify the TER score threshold which could reliably distinguish between *good* and *bad* translations according to human judgement. Following the manual evaluation, the TER threshold value was set to 0.42 for the current task. Depending on when this thresholding value is applied, we distinguish the two QE approaches used in our experiments.

3.1.1 Classification

For the classification-based approach, since training a classifier requires labelled training data, thresholding is applied on the training data prior to training in order to directly predict the two labels. For each source sentence s and its translation t' from the training corpus, we associate the label x corresponding to the rule (1):

$$x = \begin{cases} 0 & \text{if } f(t', t) > \delta \\ 1 & \text{else} \end{cases} \quad (1)$$

where t is a translation reference, f is the evaluation function (TER in our case) and δ is the determined threshold. On unseen data, the trained classifier is used to infer one of the two labels for the translation of each source sentence. In the current classification context, we associate the labels 0 and 1 with *bad* and *good* translations, respectively.

3.1.2 Regression

Unlike the classification model, the regression model can be trained on the training data without applying the threshold initially. Once the model has been built and is used to predict the scores for an unseen set of translations, the threshold value is applied to label the data set and identify *bad* translations. However, the regression approach requires the computation of 2 different threshold values: (i) a *reference threshold* set on the test set TER scores and (ii) a *prediction threshold* which is set on the TER predicted by the regression model. Setting the reference threshold to the manually set threshold value of 0.42 and using an unseen development set randomly selected from the training data, the *prediction threshold* is set by optimising the performance of the regression model with respect to an evaluation metric (precision, recall, accuracy, etc.). In the context of our experiments, the threshold is set by optimising the F1 score with label 0 as the true positive, thus optimising both precision and recall for the bad translations.

3.2 Data Selection

In order to perform data selection, we use an approach based on the technique presented by Axelrod et al. (2011), to rank out-of-domain sentence pairs according to their relevance to our target domain. According to this approach, each sentence-pair from the out-of-domain corpora is ranked according to the formula in (2):

$$[H_{i_{src}}(s) - H_{o_{src}}(s)] + [H_{i_{trg}}(s) - H_{o_{trg}}(s)] \quad (2)$$

where $H_{i_{src}}$ and $H_{o_{src}}$ refer to the cross entropy of the source sentence on the in-domain and out-of-domain language models (LM), respectively, while $H_{i_{trg}}$ and $H_{o_{trg}}$ refer to cross-entropy of the target sentences on similar target side LMs. In contrast to the ranking sentences using only target domain LM, this technique biases towards the sentences which are both *like* the in-domain corpus and *unlike* the average of the out-of-domain corpora. The out-of-domain LMs used in this context are built on a randomly selected sub-sample of the supplementary data having the same size and vocabulary as that of the in-domain LM (both for source and target). Eventually, the sentence-pairs are sorted by the scores and the lowest-scoring sentences are selected by using a threshold.

While the bilingual cross-entropy difference based approach forms the basis of our data selection technique, we use an important variation on it to suit our context: In contrast to biasing the scores towards all target-domain sentences, our approach requires bias towards the set of potentially poorly translated target domain sentences. To allow this shift of bias, the source in-domain LM is trained only on the subset of the target-domain sentences which are poorly translated by the baseline. The source-side out-of-domain LMs on the other hand, is trained on a concatenation of the remaining target-domain sentences (well translated by the baseline) and the out-of-domain corpora. Finally, we use perplexity (instead of cross-entropy) for ranking the out-of-domain sentences in our experiments.[1] Secondly, In order to make the in-domain and out-of-domain LMs comparable, we restrict the vocabulary and the size of the out-of-domain LM to that of the in-domain LM. Hence, the modified scoring function used for ranking sentences for our experiments is given as (3):

$$[PP_{i_{src'}}(s) \quad - \quad PP_{o_{src}+i_{src''}}(s)] + [PP_{i_{trg}}(s) - PP_{o_{trg}}(s)] \quad (3)$$

where $PP_{i_{src'}}$ indicates the perplexity on the in-domain LM trained only on the source-side of the poorly translated sentences while $PP_{o_{src}+i_{src''}}$ refers to the LM trained on the remaining target-domain data and out-of-domain data. Note that the target side of the scoring remains the same, as there is no notion of good or bad translations in the target side of the bitext data.

3.3 Data Combination

Multiple techniques exist in the SMT literature to combine out-of-domain data with in-domain data. The combination could be done using instance weighting (Jiang and Zhai, 2007), or by linearly interpolating the phrase tables (Foster and Kuhn, 2007). Considering the success of linear interpolation outperforming the other techniques (Sennrich, 2012), we choose this technique to combine the two datasets.

In order to learn the interpolation weights, LMs are constructed on the target side of the in-domain training set and the selected supplementary data. These LMs are then interpolated using expectation maximisation on the target side of the devset to learn the optimal mixture weights. These weights are subsequently used to combine the individual feature values for every phrase pair from two phrase-tables using a weighted linear interpolation scheme. For the LMs, individual models trained on the in-domain and selected out-of-domain datasets are interpolated in a similar fashion with interpolation weights set on the devset.

[1] As cross-entropy and perplexity are monotonically related, they produce the same ranking.

4 Experimental Setup

4.1 Datasets and Tools

The primary in-domain training data for our baseline systems comprises En–Fr bilingual datasets from Symantec TMs. Considering the wider vocabulary of the forum content, we use the freely available Europarl (EP) version 6 (Koehn, 2005) and News Commentary (NC)[2] datasets in combination with the Symantec TMs to create a stronger second baseline model. We then use the following two freely available parallel datasets from the web, as the supplementary resources for data selection experiments:

- OpenSubtitles2011 (OPS) Corpus[3].
- MultiUN (UN) Parallel Corpus[4]

	Data Set	Line Cnt.	En. Token	Fr. Token
Bi-text Data	Symantec TM	3,659,455	72,604,817	82,046,300
	Europarl	1,924,594	52,139,148	57,837,037
	News-Comm.	134,757	3,338,552	3,917,982
	Dev	1,692	22,661	25,840
	Test	1,032	13,160	15,164
Supp. Data	MultiUN	9,010,933	227,085,145	263,051,365
	Open-Subs.	19,835,265	154,307,759	145,769,773
Mono Data	English Forum	1,276,136	19,964,837	
	French Forum	83,575		908,106

Table 1: Number of sentences and token counts for training, development, test, supplementary data and forum data sets.

Monolingual Symantec forum posts in French along with the target side of the TM, EP and NC training data serve as baseline language modelling data. All the LMs in our experiments are linearly interpolated with the weights set by expectation maximisation on the development (dev) set. Furthermore, a sizeable amount of English forum data (Banerjee et al., 2012) is used to create the source-side target-domain LM which is used both in the

[2] http://www.statmt.org/wmt12/translation-task.html
[3] http://www.opensubtitles.org/
[4] http://www.euromatrixplus.net/multi-un/

data selection and in determining the set of potentially poorly translated sentences in the target-domain. The dev and test sets are randomly selected from this English forum data and manually translated by professional translators. Table 1 reports the statistics on all the datasets used in all our experiments.

The SMT system used in our experiments is based on the standard phrase-based SMT toolkit: Moses (Koehn et al., 2007). The feature weights are tuned using Minimum Error Rate Training (Och, 2003) on the devset. All the LMs in our experiments are created using the IRSTLM (Federico et al., 2008) language modelling toolkit. Finally, translations of the test sets in every phase of our experiments are evaluated using BLEU, METEOR (Banerjee and Lavie, 2005) and TER (Snover et al., 2006) scores.

The classification and regression models used in the QE component of our approach are based on Support Vector Machines (SVMs) (Joachims, 1999) using Radial Basis function (RBF) kernels. We use the LibSVM toolkit:[5] a free open source implementation of the technology, for all our classification/regression model training and predictions. In order to tune the features of the SVM-based classification and regression models the grid search functionality associated with LibSVM is used. The process of feature extraction is performed using an inhouse tool.

4.2 QE Results

As stated in Section 3.1, we use both the classification and regression approaches to the QE task. For both models, we use 1200 randomly selected sentences from the devset (Table 1), to actually train the model and the remaining 492 sentences to optimise the SVM parameters using grid search. The classification and regression models are both evaluated on the available testset.

For the classification-based QE, we label the training data sentences by using the manually set threshold (0.42) on their TER scores. The testset is also labelled likewise. Once the SVM parameters have been set and the model has been trained, it is used to classify the testset and the resulting predictions are compared to that of the reference predictions. For the regression setup, the model is trained using the training data and associated TER scores, and this model is used to predict the TER scores on the testset. Comparing the predicted TER scores with the true TER scores for the testset, helps us predict the performance of the regression model in terms of root-mean-squared-error (RMSE) and minimum-average-error (MAE). However once the predictions are achieved, both the predictions and the reference TERs are converted to class-label representations by applying the *prediction* and *reference thresholds*, respectively. This allows us to compare the effect of the regression approach in terms of the same metrics (F1 score) used to evaluate the classifier-based approach. For the regression setup, the prediction threshold is set by optimising the F1 score on the regression-model predictions on the devset. Figure 1 shows the variation of F1 scores for different values of the prediction threshold, and our choice of threshold value of 0.4 corresponding to the best F1 score.

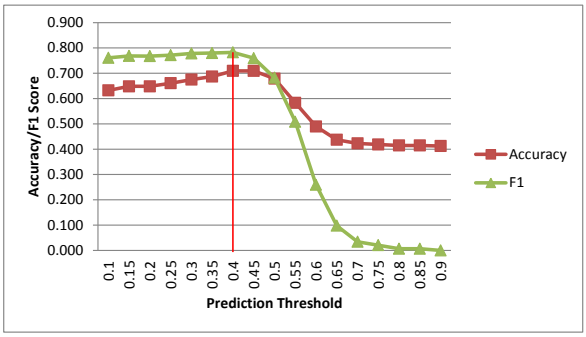


Figure 1: Variation of F1 score with prediction threshold on devset for Regression Setup.

Table 2 presents the F1 score and accuracy results on the testset for both the classification and regression setups. The accuracy and F1 scores for the regression setup correspond to an RMSE value of 0.2899 and an MAE value of 0.2104. The final column in the table indicates the percentage of the English forum data labelled as *bad* translations by the QE setup.

Configuration	Accuracy	F1 Score	% on Forum
Classification	75.2	0.8028	83.2
Regression	72.8	0.7860	83.4

Table 2: Accuracy and F1 scores on testset using classification and MAE and RMSE using regression on testset.

The results in Table 2 clearly show that using binary classification we achieve a higher accuracy on the QE task. Hence we use this particular configuration as the choice of our QE approach in order to identify potentially bad translations for data

[5]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

selection. This corresponds to 83.4% (1,062,243 sentences) of the forum sentences being labelled as potentially badly translated.

4.3 SMT Results

In order to compare the effect of our approach to that of more conventional approaches, we conduct experiments on the following 5 models:

1. **BL1**: A baseline SMT model trained only on Symantec TMs.
2. **BL2**: A baseline SMT model trained on concatenated data from Symantec TMs, EP and NC parallel data sets.
3. **Full**: Using the entire supplementary datasets (either OPS or UN) in combination with the baseline (BL2).
4. **PPD**: Selecting supplementary data for the baseline (BL2) using the entire target-domain as the reference set with bilingual difference of cross-entropy (Axelrod et al., 2011).
5. **QESel**: Using our proposed approach of data selection by modified bilingual difference of perplexity (Equation 3).

We use two baseline configurations, where *BL1* is trained only on Symantec TMs while *BL2* uses additional (out-of-domain) parallel data to address data sparseness issues in the in-domain corpus. After ranking the supplementary sentence pairs using the *PPD* and *QESel* approaches, we need a threshold value to select only a section of the selected data for adaptation. In order to compare the relative effects of the two approaches on the same amount of data, we used 6 different threshold values approximately aimed at 10%, 20%, 30%, 40%, 50% and 60% of the entire datasets.

The individual translation and language models trained on these selected datasets are finally combined with the baseline models using linear interpolation techniques detailed in Section 3.3. Table 3 presents the BLEU, METEOR and TER scores for all the different configurations used in our experiments. For the *PPD* and *QESel* configurations, we present the scores for all the six sub-configurations corresponding to different sizes of the selected data. Best scores for the *QESel* and *PPD* configurations are in bold, with $*$ and † representing statistical significance over the baseline (BL2) and *Full* configurations, respectively.

The two baseline scores in Table 3 clearly indicate that *BL2* is a stronger baseline with the improvements over *BL1* being statistically significant at the p=0.05 level using bootstrap resampling (Koehn, 2004). Hence, the subsequent models are evaluated with respect to the stronger baseline (*BL2*) scores. The results show that using the supplementary data even without data selection (*Full* configuration) improves the translation quality scores. Using the UN as the supplementary data we observe a gain of 0.36 absolute BLEU points while the gain is 0.67 absolute when using OPS as the supplementary source. While the gain from using OPS as supplementary data source is statistically significant, the improvement provided by the UN datasets is not significant.

Config.		UN			OPS		
		BLEU	METEOR	TER	BLEU	METEOR	TER
BL1		31.15	48.47	0.5636	31.15	48.47	0.5636
BL2		32.27	50.19	0.5551	32.27	50.19	0.5551
Full		32.63	49.92	0.5518	*32.94	50.06	0.5460
PPD	10%	*32.75	50.03	0.5516	*32.90	50.27	0.5524
	20%	32.59	50.19	0.5518	*33.06	50.31	0.5460
	30%	*32.87	49.93	0.5473	*33.25	50.45	0.5446
	40%	*32.93	50.11	0.5489	*33.13	50.43	0.5460
	50%	*†**33.07**	50.18	0.5432	*†**33.52**	50.55	0.5450
	60%	32.59	49.93	0.5520	*33.06	50.32	0.5463
QESel	10%	*32.86	50.14	0.5458	*33.08	50.38	0.5448
	20%	*32.88	50.16	0.5487	*†**33.59**	50.96	0.5360
	30%	*†**33.19**	50.41	0.5383	*†33.46	50.63	0.5391
	40%	*†33.13	50.24	0.5451	*†33.39	50.49	0.5442
	50%	*32.84	50.21	0.5456	*†33.53	50.70	0.5451
	60%	*32.79	50.24	0.5489	*33.21	50.53	0.5448

Table 3: Testset BLEU, METEOR and TER scores for the different data selection configurations.

Using the *PPD* approach, we observe an improvement over the *BL2* baseline and the *Full* configuration using only a fraction of the datasets in most cases. For UN, using 50% of the data, we observe improvements of 0.8 and 0.44 absolute BLEU points over the baseline and *Full* configurations, respectively. The improvement figures are 1.25 and 0.58 absolute BLEU points over the baseline and Full configurations, respectively, using only 50% of the OPS dataset. All these improvements are statistically significant. METEOR and TER also follow a similar trend of improvement compared to BLEU.

The *QESel* approach, also provides statistically significant improvements over the *BL2* baseline for all sections of the full datasets. Again scores improve significantly over the *Full* configuration for most of the fraction of datasets used in our experiments. We observe an improvement of 0.92 and 0.56 absolute BLEU points using only 30% of the

UN data over the *BL2* baseline and *Full* scores, respectively. Using 20% of the entire OPS dataset, we observe improvements of 1.32 and 0.65 absolute BLEU points over the *BL2* baseline and *Full* scores, respectively. All these improvements are statistically significant at the p=0.05 level. The other evaluation metric scores also follow a similar trend of improvements. The *QESel* approach is also observed to consistently outperform the corresponding *PPD* scores for similar sizes of the supplementary datasets (the only exception being the 50% scores for UN).

5 Discussion

Comparing the improvements obtained by the two data selection approaches in Section 4.3, we observe that the *QESel* method achieves the best scores using significantly smaller amounts of data compared to the PPD approach. The *QESel* approach achieves the best improvements with only 30% and 20% of the supplementary data, for UN and OPS datasets, respectively, compared to the 50% data selected by the *PPD* approach. Furthermore, this approach provides scores which are consistently higher than the corresponding *PPD* approach for the same amount of selected sentences. Since the *QESel* approach is driven only by the poorly translated sentences in the target-domain, it prioritises the supplementary sentence pairs relevant to them. In contrast, the *PPD* approach has no particular preference towards such supplementary sentence pairs. As a consequence, selecting the top sentence pairs using the *QESel* approach improves only the previously poorly translated sentences, while *PPD* aims at uniformly improving all the target-domain sentences in general. This difference causes the *QESel* approach to achieve higher translation scores with lesser amounts of data in the current context.

To further illustrate our point, in Table 4 we present two example sentences from our testset whose *BL2* translations are labelled *good* and *bad* by the QE classifier, along with their *PPD* and *QESel* translations. The first example shows that the *PPD* approach leads to a better syntax compared to the baseline and the *QESel* approach by ordering *Pouvez-vous* properly for an interrogative sentence. Also, the verb *permettre* is in its infinitive form which is correct in this context, while the same verb is wrong in the *QESel* translations.

Good Translation	SRC	Re : Can you make it possible for users to delete their account ?
	REF	Re : Pouvez-vous accorder aux utilisateurs le droit de supprimer leur compte ?
	BL2	Re : Vous pouvez vous permettent aux utilisateurs de supprimer son compte ?
	PPD	Re : **Pouvez-vous** vous **permettre** aux utilisateurs de supprimer son compte ?
	QESel	Re : Est-ce que vous permettent aux utilisateurs de supprimer son compte ?
Bad Translation	SRC	Looks to me like the " Restart " button is highlighted - and I had just restarted (not done any update) .
	REF	Il me semble que le bouton " Redémarrer " est en surbrillance et je venais juste de redémarrer (et non d' effectuer des mises à jour) .
	BL2	On dirait que le " Redémarrer " bouton est mis en évidence - et j' ai redémarré (pas fait une mise à jour) .
	PPD	Il me semble que le " Redémarrer " bouton est mis en évidence - et j' ai redémarré (pas fait une mise à jour) .
	QESel	Il me semble que **le bouton " Redémarrer "** est mis en évidence - et j' **avais** redémarré (pas fait une mise à jour) .

Table 4: Example sentences and their translations.

This example shows how for sentences with decent baseline translation, the *PPD* approach performs better than the *QESel* approach. The second example on the other hand, shows that the *QESel* method leads to a better word ordering and keeps the correct past tense for the verb *had restarted* translated as *avais redémarré*, in comparison to *PPD* translations. The *BL2* translation for the example being *bad*, the focussed data selection by *QESel* improves it further than the conventional *PPD* approach.

Furthermore, our experiments reveal that the OPS datasets provide better improvements using both data selection methods in contrast to the UN corpus. This may be due to the informal and colloquial nature of the OPS corpus which makes it more appropriate to adapting SMT models for translating forum content.

6 Conclusion and Future Work

In this paper we presented a QE-guided data selection approach for domain adaptation of SMT systems using only the part of the target-domain data that is poorly translated by the baseline system. Our experiments revealed that this approach performs significantly better than the unadapted baseline model as well as the model using the entire supplementary data without any data selection. Furthermore, this approach also achieves similar or better improvements to that of conventional data selection approaches with considerably smaller amounts of selected data.

Despite using a set of baseline features for the QE task, our approach shows promising results thereby indicating a number of possible future directions. Extending the set of QE features to im-

prove prediction/classification performance is the primary future direction. We would also like to investigate the effect of this approach using a finer grained classification approach. Finally a deeper investigation into sophisticated data selection and ranking schemes is necessary to further exploit the effectiveness of the approach.

Acknowledgments

This work is supported by the European Commission's Seventh Framework Programme (Grant 288769) from Science Foundation Ireland (Grant 07/CE/I1142) as part of CNGL at Dublin City University, and from Research Ireland under the Enterprise Partnership Scheme (EPSPD/2011/135).

References

Axelrod, A., X. He, and J. Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on EMNLP-11*, pages 355–362, Edinburgh, United Kingdom.

Banerjee, S. and A. Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI.

Banerjee, P., S. Naskar, J. Roturier, A. Way, and J. van Genabith. 2012. Translation Quality-Based Supplementary Data Selection by Incremental Update of Translation Models. In *Proceedings of COLING-2012*, pages 149–165, Mumbai, India.

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2003. Confidence Estimation for Machine Translation. In *JHU/CLSP Summer Workshop Final Report*.

Callison-Burch, C., P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51.

Federico, M., N. Bertoldi, and M. Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Interspeech 2008*, pages 1618–1621, Brisbane, Australia.

Foster, G. and R. Kuhn. 2007. Mixture-model adaptation for SMT. In *ACL 2007: Proceedings of the Second WMT*, pages 128–135, Prague, Czech Republic.

Hildebrand, A. S., M. Eck, S. Vogel, and A. Waibel. 2005. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. In *Proceedings of 10th EAMT Conference*, pages 119–125, Budapest, Hungary.

Jiang, J. and C. Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of ACL*, pages 264–271, Prague, Czech Republic.

Joachims, T. 1999. Making Large-Scale SVM Learning Practical. In Schölkopf, B., C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.

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

Koehn, P. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Conference on EMNLP, (EMNLP 2004)*, pages 388–395, Barcelona, Spain.

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

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

Okita, T., R. Rubino, and J. van Genabith. 2012. Sentence-Level Quality Estimation for MT System Combination. In *Proceedings of the ML4HMT-12 Workshop*, page 55.

Rubino, R., S. Huet, F. Lefèvre, and G. Linarès. 2012. Statistical post-editing of machine translation for domain adaptation. *Proceedings of the European Association for Machine Translation (EAMT)*, pages 221–228.

Sennrich, R. 2012. Mixture-Modeling with Unsupervised Clusters for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 16th Annual Conference of the EAMT (EAMT-2012)*, pages 185–192, Trento, Italy.

Snover, M., B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*, pages 223–231, Cambridge, MA.

Specia, L., N. Hajlaoui, C. Hallett, and W. Aziz. 2011. Predicting machine translation adequacy. *Proceedings of MT Summit XIII*, pages 19–23.

Ueffing, N., K. Macherey, and H. Ney. 2003. Confidence Measures for Statistical Machine Translation. In *Proceedings of the MT Summit IX*.

Two Approaches to Correcting Homophone Confusions in a Hybrid Machine Translation System

Pierrette Bouillon[1], **Johanna Gerlach**[1], **Ulrich Germann**[2], **Barry Haddow**[2], **Manny Rayner**[1]
(1) FTI/TIM, University of Geneva, Switzerland
{Pierrette.Bouillon,Johanna.Gerlach,Emmanuel.Rayner}@unige.ch
(2) School of Informatics, University of Edinburgh, Scotland
{ugermann,bhaddow}@inf.ed.ac.uk

Abstract

In the context of a hybrid French-to-English SMT system for translating online forum posts, we present two methods for addressing the common problem of homophone confusions in colloquial written language. The first is based on hand-coded rules; the second on weighted graphs derived from a large-scale pronunciation resource, with weights trained from a small bicorpus of domain language. With automatic evaluation, the weighted graph method yields an improvement of about $+0.63$ BLEU points, while the rule-based method scores about the same as the baseline. On contrastive manual evaluation, both methods give highly significant improvements ($p < 0.0001$) and score about equally when compared against each other.

1 Introduction and motivation

The data used to train Statistical Machine Translation (SMT) systems is most often taken from the proceedings of large multilingual organisations, the generic example being the Europarl corpus (Koehn, 2005); for academic evaluation exercises, the test data may well also be taken from the same source. Texts of this kind are carefully cleaned-up formal language. However, real MT systems often need to handle text from very different genres, which as usual causes problems.

This paper addresses a problem common in domains containing informally written text: spelling errors based on homophone confusions. Concretely, the work reported was carried out in the context of the ACCEPT project, which deals with the increasingly important topic of translating online forum posts; the experiments we describe were performed using French data taken from the Symantec forum, the concrete task being to translate it into English. The language in these posts is very far from that which appears in Hansard. People write quickly and carelessly, and no attempt is made to clean up the results. In particular, spelling is often uncertain.

One of the particular challenges in the task considered here is that French has a high frequency of homophones, which often cause confusion in written language. Everyone who speaks English is familiar with the fact that careless writers may confuse *its* ("of or belonging to it") and *it's* (contraction of "it is" or "it has"). French has the same problem, but to a much greater degree. Even when someone is working in an environment where an online spell-checker is available, it is easy to write *ou* ("or") instead of *où* ("where"), *la* ("the-feminine") instead of *là* ("there") or *ce* ("this") instead of *se* ("him/herself"). Even worse, there is systematic homophony in verb-form endings: for example, *utiliser* ("to use") *utilisez* ("you use") and *utilisé* ("used") are all homophones.

In French posts from the Symantec forum, we find that between 10% and 15% of all sentences contain at least one homophone error, depending on exactly how the term is defined[1]. Substituting a word with an incorrect homophone will often result in a translation error. Figure 1 shows typical examples of homophone errors and their effect on translation.

The core translation engine in our application is a normal SMT system, bracketed between pre- and post-editing phases. In what follows, we contrast two different approaches to handling homophone errors, which involve pre-editing in different ways. The first approach is based on knowledge-intensive construction of regular expression rules, which use the surrounding context to correct the most frequent types of homophone

[1] Unclear cases include hyphenation, elison and some examples of missing or incorrect accents.

	source	automatic translation
original	**La sa** ne pose pas de problème ...	**The its** is not the issue ...
corrected	**Là ça** ne pose pas de problème ...	**Here it** is not a problem
original	... (du moins on ne **recoit** pas l'alerte).	... (at least **we do not recoit** alert).
corrected	... (du moins on ne **reçoit** pas l'alerte).	.. (at least **it does not receive** the alert).

Figure 1: Examples of homophone errors in French forum data, contrasting English translations produced by the SMT engine from plain and corrected versions.

confusions.

The second is an engineering method: we use a commercial pronunciation-generation tool to generate a homophone dictionary, then use this dictionary to turn the input into a weighted graph where each word is replaced by a weighted disjunction of homophones. Related, though less elaborate, work has been reported by Bertoldi et al. (2010), who address spelling errors using a character-level confusion network based on common character confusions in typed English and test them on artificially created noisy data. Formiga and Fonollosa (2012) also used character-based models to correct spelling on informally written English data.

The two approaches in the present paper exploit fundamentally different knowledge sources in trying to identify and correct homophone errors. The rule-based method relies exclusively on source-side information, encoding patterns indicative of common French homophone confusions. The weighted graph method shifts the balance to the target side; the choice between potential homophone alternatives is made primarily by the target language model, though the source language weights and the translation model are also involved.

The rest of the paper is organised as follows. Section 2 describes the basic framework in more detail, and Section 3 the experiments. Section 4 summarises and concludes.

2 Basic framework

The goal of the ACCEPT project is to provide easy cross-lingual access to posts in online forums. Given the large variety of possible technical topics and the limited supply of online gurus, it frequently happens that users, searching forum posts online, find that the answer they need is in a language they do not know.

Currently available tools, for example Google Translate, are of course a great deal better than nothing, but still leave much to be desired. When one considers that advice given in an online forum may not be easy to follow even for native language speakers, it is unsurprising that a Google-translated version often fails to be useful. There is consequently strong motivation to develop an infrastructure explicitly designed to produce high-quality translations. ACCEPT intends to achieve this by a combination of three technologies: pre-editing of the source; domain-tuned SMT; and post-editing of the target. The pre- and post-editing stages are performed partly using automatic tools, and partly by manual intervention on the part of the user communities which typically grow up around online forums. We now briefly describe the automatic parts of the system.

2.1 SMT engine and corpus data

The SMT engine used is a phrase-based system trained with the standard *Moses* pipeline (Koehn et al., 2007), using GIZA++ (Och and Ney, 2000) for word alignment and SRILM (Stolcke, 2002) for the estimation of 5-gram Kneser-Ney smoothed (Kneser and Ney, 1995) language models.

For training the translation and lexicalised reordering models we used the releases of europarl and news-commentary provided for the WMT12 shared task (Callison-Burch et al., 2012), together with a dataset from the ACCEPT project consisting mainly of technical product manuals and marketing materials.

For language modelling we used the target sides of all the parallel data, together with approximately 900 000 words of monolingual English data extracted from web forums of the type that we wish to translate. Separate language models were trained on each of the data sets, then these were linearly interpolated using SRILM to minimise perplexity on a heldout portion of the forum data.

For tuning and testing, we extracted 1022 sentences randomly from a collection of monolingual French Symantec forum data (distinct from the monolingual English forum data), translated these using Google Translate, then post-edited to create references. The post-editing was performed by a native English speaker, who is also fluent in French. This 1022-sentence parallel text was then split into two equal halves (`devtest_a` and `devtest_b`) for minimum error rate tuning (MERT) and testing, respectively.

2.2 Rule-based pre-editing engine

Rule-based processing is carried out using the Acrolinx engine (Bredenkamp et al., 2000), which supports spelling, grammar, style and terminology checking. These methods of pre-editing were originally designed to be applied by authors during the technical documentation authoring process. The author gets error markings and improvement suggestions, and decides about reformulations. It is also possible to apply the provided suggestions automatically as direct reformulations. Rules are written in a regular-expression-based formalism which can access tagger-generated part-of-speech information. The rule-writer can specify both positive evidence (patterns that will trigger application of the rule) and negative evidence (patterns that will block application).

3 Experiments

We compared the rule-based and weighted graph approaches, evaluating each of them on the 511 sentence `devtest_b` corpus. The baseline SMT system, with no pre-editing, achieves an average BLEU score of 42.47 on this set.

3.1 The rule-based approach

Under the ACCEPT project, a set of lightweight pre-editing rules have been developed specifically for the Symantec Forum translation task. Some of the rules are automatic (direct reformulations); others present the user with a set of suggestions. The evaluations described in Gerlach et al. (2013) demonstrate that pre-editing with the rules has a significant positive effect on the quality of SMT-based translation.

The implemented rules address four main phenomena: differences between informal and formal language (Rayner et al., 2012), differences between local French and English word-order, elision/punctuation, and word confusions. Rules for resolving homophone confusions belong to the fourth group. They are shown in Table 1, together with approximate frequencies of occurrence in the development corpus.

Table 1: Hand-coded rules for homophone confusions and per-sentence frequency of applicability in the development corpus. Some of the rules also cover non-homophone errors, so the frequency figures are slight overestimates as far as homophones are concerned.

Rule	Freq.
a/as/à	4.17%
noun phrase agreement	3.20%
incorrect verb ending (er/é/ez)	2.90%
missing hyphenation	2.08%
subject verb agreement	1.90%
missing elision	1.26%
du/dû	0.35%
la/là	0.32%
ou/où	0.28%
ce/se	0.27%
Verb/noun	0.23%
tous/tout	0.22%
indicative/imperative	0.19%
future/conditional tense	0.14%
sur/sûr	0.10%
quel que/quelque	0.08%
ma/m'a	0.06%
quelle/qu'elle/quel/quels	0.05%
ça/sa	0.04%
des/dès	0.04%
et/est	0.02%
ci/si	0.01%
m'y/mi/mis	0.01%
other	0.17%
Total	18.09%

The set of Acrolinx pre-editing rules potentially relevant to resolution of homophone errors was applied to the `devtest_b` set test corpus (Section 2.1). In order to be able to make a fair comparison with the weighted-graph method, we only used rules with a unique suggestion, which could be run automatically. Applying these rules produced 430 changed words in the test corpus, but did not change the average BLEU score significantly (42.38).

Corrections made with a human in the loop, used as "oracle" input for the SMT system, by the

way, achieve an average BLEU score[2] of 43.11 — roughly on par with the weighted-graph approach described below.

3.2 The weighted graph approach

In our second approach, the basic idea is to transform the input sentence into a *confusion network* (Bertoldi et al., 2008) which presents the translation system with a weighted list of homophone alternatives for each input word. The system is free to choose a path through a network of words that optimizes the internal hypothesis score; the weighting scheme for the alternatives can be used to guide the decoder. The conjecture is that the combination of the confusion network weights, the translation model and the target language model can resolve homophone confusions.

3.2.1 Defining sets of confusable words

To compile lists of homophones, we used the commercial Nuance Toolkit `pronounce` utility as our source of French pronunciation information.

We began by extracting a list of all the lexical items which occurred in the training portion of the French Symantec forum data, giving us 30 565 words. We then ran `pronounce` over this list. The Nuance utility does not simply perform table lookups, but is capable of creating pronunciations on the fly; it could in particular assign plausible pronunciations to most of the misspellings that occurred in the corpus. In general, a word is given more than one possible pronunciation. This can be for several reasons; in particular, some sounds in French can systematically be pronounced in more than one way, and pronunciation is often also dependent on whether the word is followed by a consonant or vowel. Table 2 shows examples.

Using the data taken from `pronounce`, we grouped words together into clusters which have a common pronunciation; since words typically have more than one pronunciation, they will typically also belong to more than one cluster. We then contructed sets of possible alternatives for words by including, for each word W, all the words W' such that W and W' occurred in the same cluster; since careless French writing is also characterised by mistakes in placing accents, we added all words W' such that W and W' are identical up to dropping accents. Table 3 shows typical results.

[2]With parameter sets from tuning the system on raw input and input preprocessed with the fully automatic rules; cf. Sec. 3.3.

Word	Pronunciation
ans	A~ A~z
prévu	p r E v y p r e v y
québec	k e b E k
roule	r u l r u l *

Table 2: Examples of French pronunciations generated by `pronounce`. The format used is the Nuance version of ARPABET.

Intuitively, it is in general unlikely that, on seeing a word which occurs frequently in the corpus, we will want to hypothesize that it may be a misspelling of one which occurs very infrequently. We consequently filtered the sets of alternatives to remove all words on the right whose frequency was less than 0.05 times that of the word on the left.

Table 3: Examples of sets of possible alternatives for words, generated by considering both homophone and accent confusions.

Word	Alternatives
aux	au aux haut
créer	créer créez créé créée créées créés
côte	cote coté côte côté quot quote
hôte	haut haute hôte hôtes
il	e elle elles il ils l le y
mène	main mené mène
nom	nom noms non
ou	ou où
saine	sain saine saines scène seine
traits	trait traits tray tre tres très

3.2.2 Setting confusion network weights

In a small series of preliminary experiments we first tested three naïve weighting schemes for the confusion networks.

- using a **uniform** distribution that assigns equal weight to all spelling alternatives;
- setting weights proportional to the **unigram probability** of the word in question;
- computing the weights as state probabilities in a trellis with the **forward-backward** algorithm (Rabiner, 1989), an algorithm widely

Table 4: Decoder performance with different confusion network weighting schemes.

weighting scheme	av. BLEU[a]	std.
none (baseline system)	42.47	± .22
uniform	41.50	± .37
unigram	41.58	± .26
fwd-bwd (bigram)	41.81	± .16
bigram context (interpolated)	43.10	± .32

[a]Based on muliple tuning runs with random parameter initializations.

used in speech recognition. Suppose that each word $\hat{w}_i$ in the observed translation input sentence is produced while the writer has a particular "true" word $w_i \in C_i$ in mind, where C_i is the set of words confusable with $\hat{w}_i$. For the sake of simplicity, we assume that within a confusion set, all "true word" options are equally likely, i.e., $p(\hat{w}_i \mid w_i = x) = \frac{1}{|C_i|}$ for $x \in C_i$. The writer chooses the next word w_{i+1} according to the conditional word bigram probability $p(w_{i+1} \mid w_i)$.

The *forward* probability $fwd_i(x)$ is the probability of arriving in state $w_i = x$ at time i, regardless of the sequence of states visited en-route; the backward probability $bwd_i(x)$ is the probability of arriving at the end of the sentence coming from state $w_i = x$, regardless of the path taken. These probabilities can be computed efficiently with dynamic programming.

The weight assigned to a particular homophone alternative x at position i in the confusion network is the joint forward and backward probability:

$$weight_i(x) = fwd_i(x) \cdot bwd_i(x).$$

In practice, it turns out that these three naïve weighting schemes do more harm than good, as the results in Table 4 show. Clearly, they rely too much on overall language statistics (unigram and bigram probabilities) and pay too little attention to the actual input.

We therefore designed a fourth weighting scheme ("**bigram context interpolated**") that gives more weight to the observed input and computes the weights as the average of two score components. The first is a binary feature function that assigns 1 to each word actually observed in the input, and 0 to its homophone alternatives. The second component is the bigram-based in-context probability of each candidate. Unlike the forward-backward weighting scheme, which considers all possible context words for each candidate (as specified in the respective confusion sets), the new scheme only considers the words in the actual input as context words.

It would have be desirable to keep the two score components separate and tune their weights together with all the other parameters of the SMT system. Unfortunately, the current implementation of confusion network-based decoding in the *Moses* decoder allows only one single weight in the specification of confusion networks, so that we had to combine the two components into one score before feeding the confusion network into the decoder.

With the improved weighting scheme, the confusion network approach does outperform the baseline system, giving an average BLEU of 43.10 ($+0.63$).

3.3 Automatic evaluation (BLEU)

Due to the relatively small size of the evaluation set and instability inherent in minimum error rate training (Foster and Kuhn, 2009; Clark et al., 2011), results of *individual* tuning and evaluation runs can be unreliable. We therefore preformed multiple tuning and evaluation runs for each system (baseline, rule-based and weighted graph). To illustrate the precision of the BLEU score on our data sets, we plot in Fig. 2 for each individual tuning run the BLEU score achieved on the tuning set (x-axis) against the performance on the evaluation set (y-axis). The variance along the x-axis for each system is due to search errors in parameter optimization. Since the search space is not convex, the tuning process can get stuck in local maxima. The apparent poor local correlation between performance on the tuning set and performance on the evaluation set for each system shows the effect of the sampling error.

With larger tuning and evaluation sets, we would expect the correlation between the two to improve. The scatter plot suggests that the weighted-graph system does on average produce significantly better translations (with respect to BLEU) than both the baseline and the rule-based system, whereas the difference between the baseline and the rule-based system is within the range

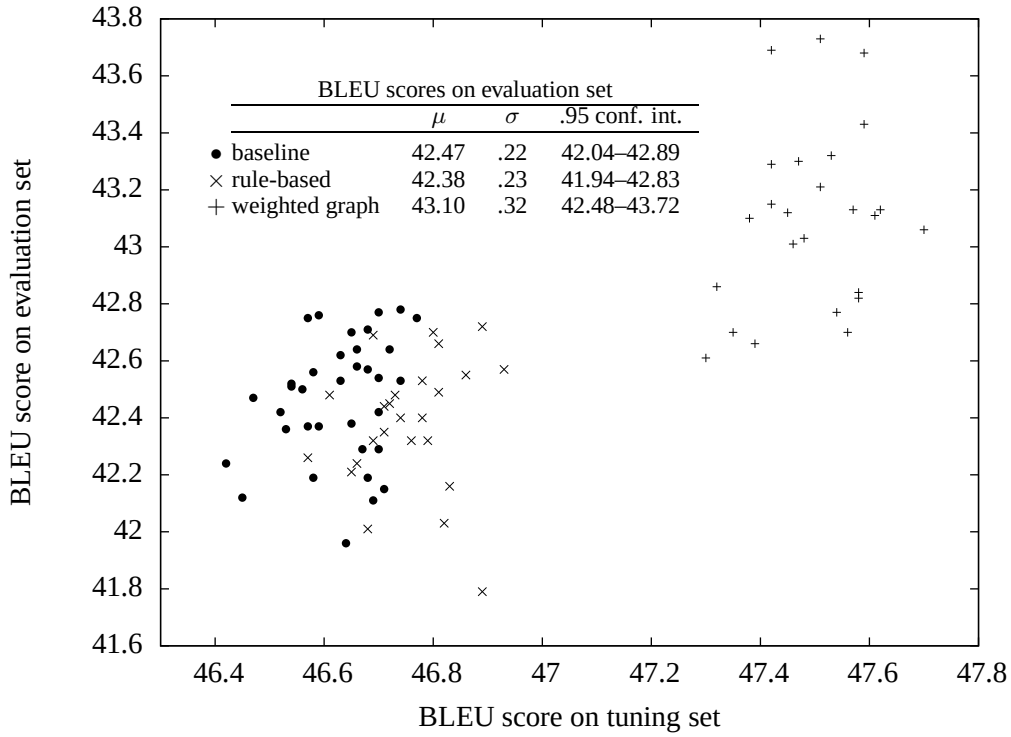


Figure 2: BLEU scores (in points) for the baseline, rule-based and weighted graph-based systems.

of statistical error.

To study the effect of tuning condition (tuning on raw vs. input pre-processed by rules), we also translated both the raw and the pre-processed evaluation corpus with all parameter setting that we had obtained during the various experiments. Figure 3 plots (with solid markers) performance on raw input (x-axis) against translation of pre-processed input (y-axis). We observe that while preprocessing harms performance for certain parameter settings, most of the time proprocessing does lead to improvements in BLEU score. The slight deterioration we observed when comparing system tuned on exactly the type of input that they were to translate later (i.e., raw or preprocessed) seems to be a imprecision in the measurement caused by training instability and sampling error rather than the result of systematic input deterioration due to preprocessing. Overall, the improvements are small and not statistically significant, but there appears to be a positive trend.

To gauge the benefits of more extensive preprocessing and input error correction we produced and translated 'oracle' input by also applying rules from the Acrolinx engine that currently require a human in the loop who decides whether or not the rule in question should be applied. The boost in performance is shown by the hollow markers in Fig. 3. Here, translation of pre-processed input consistently fares better than translation of the raw input.

3.4 Human evaluation

Although BLEU suggests that the weighted-graph method significantly outscores both the baseline and the rule-based method ($p < 0.05$ over 25 tuning runs), the absolute differences are small, and we decided that it would be prudent to carry out a human evaluation as well. Following the methodology of Rayner et al. (2012), we performed contrastive judging on the Amazon Mechanical Turk (AMT) to compare different versions of the system. Subjects were recruited from Canada, a bilingual French/English country, requesting English native speakers with good written French; we also limited the call to AMT workers who had already completed at least 50 assignments, at least 80% of which had been accepted. Judging assignments were split into groups of 20 triplets, where each triplet consisted of a source sentence and two different target sentences; the judge was asked to say which translation was better, using a five-point scale {better, slightly-better, about-equal, slightly-worse, worse}. The order of the two targets was

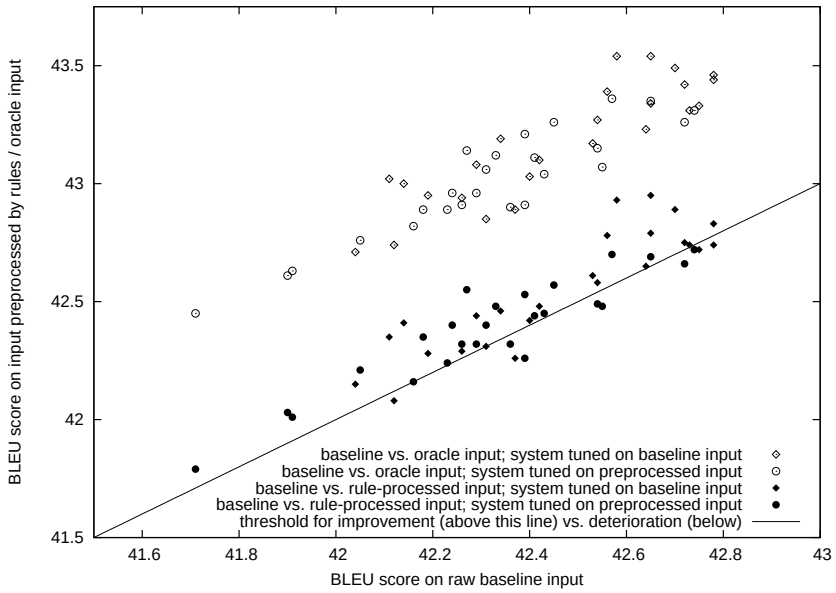


Figure 3: BLEU scores (in points) the two input conditions "baseline" and "rule-based" (solid markers). The hollow markers show the BLEU score on human-corrected 'oracle' input using a more extensive set of rules / suggestions from the Acrolinx engine that require a human in the loop.

randomised. Judges were paid $1 for each group of 20 triplets. Each triplet was judged three times.

Using the above method, we posted AMT tasks to compare a) the baseline system against the rule-based system, b) the baseline system against the best weighted-graph system (**interpolated-bigram**) from Section 3.2.2 and c) the rule-based system and the weighted-graph system against each other. The results are shown in Table 5; in the second and third columns, disagreements are resolved by majority voting, and in the fourth and fifth we only count cases where the judges are unanimous, the others being scored as unclear. In both cases, we reduce the original five-point scale to a three-point scale {better, equal/unclear, worse}[3]. Irrespective of the method used to resolve disagreements, the differences "rule-based system/baseline" and "weighted-graph system/baseline" are highly significant ($p < 0.0001$) according to the McNemar sign test, while the difference "rule-based system/weighted-graph system" is not significant.

Table 5: Comparison between baseline, rule-based and weighted-graph versions, evaluated on the 511-utterance `devtest_b` corpus and judged by three AMT-recruited judges. Figures are presented both for majority voting and for unanimous decisions only.

	Majority		Unanimous	
baseline vs rule-based				
baseline better	83	16.2%	48	9.4%
r-based better	204	40.0%	161	31.5%
Unclear	36	7.0%	93	18.1%
Equal	188	36.8%	209	40.9%
baseline vs weighted-graph				
baseline better	115	22.5%	52	10.1%
w-graph better	193	37.8%	119	23.3%
Unclear	46	9.0%	99	19.4%
Equal	157	30.7%	241	47.2%
rule-based vs weighted-graph				
r-based better	141	27.6%	68	13.3%
w-graph better	123	24.1%	70	13.7%
Unclear	25	4.9%	142	27.8%
Equal	222	43.4%	231	45.2%

We were somewhat puzzled that BLEU makes the weighted-graph system clearly better than the rule-based one, while manual evaluation rates them as approximately equal. The explanation seems to be to do with the fact that manual evaluation operates at the sentence level, giving equal importance to all sentences, while BLEU oper-

[3]For reasons we do not fully understand, we get better inter-judge agreement this way than we do when we originally ask for judgements on a three-point scale.

ates at the word level and consequently counts longer sentences as more important. If we calculate BLEU on a per-sentence basis and then average the scores, we find that the results for the two systems are nearly the same; per-sentence BLEU differences also correlate reasonably well with majority judgements (Pearson correlation coefficient of 0.39). It is unclear to us, however, whether the difference between per-sentence and per-word BLEU evaluation points to anything particularly interesting.

4 Conclusions

We have presented two methods for addressing the common problem of homophone confusions in colloquial written language in the context of an SMT system. The weighted-graph method produced a small but significant increase in BLEU, while the rule-based one was about the same as the baseline. Both methods, however, gave clearly significant improvements on contrastive manual evaluation carried out through AMT, with no significant difference in performance when the two were compared directly.

The small but consistent improvements in BLEU score that we observed with the human-in-the-loop oracle input over the fully automatic rule-based setup invite further investigation. How many of the decisions currently left to the human can be automated? Is there a fair way of comparing and evaluating fully automatic against semi-automatic setups? Work on these topics is in preparation and will be reported elsewhere.

Acknowledgements

The work described in this paper was performed as part of the Seventh Framework Programme ACCEPT project, under grant agreement 288769.

References

Bertoldi, Nicola, Mauro Cettolo, and Marcello Federico. 2010. “Statistical machine translation of texts with misspelled words.” *NAACL*. Los Angeles, CA, USA.

Bertoldi, Nicola, Richard Zens, Marcello Federico, and Wade Shen. 2008. “Efficient speech translation through confusion network decoding.” *IEEE Transactions on Audio, Speech & Language Processing*, 16(8):1696–1705.

Bredenkamp, Andrew, Berthold Crysmann, and Mirela Petrea. 2000. “Looking for errors : A declarative formalism for resource-adaptive language checking.” *LREC*. Athens, Greece.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, et al. (eds.). 2012. *Seventh Workshop on Statistical Machine Translation (WMT)*. Montréal, Canada.

Clark, Jonathan H., Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. “Better hypothesis testing for statistical machine translation: Controlling for optimizer instability.” *ACL-HLT*. Portland, OR, USA.

Formiga, Lluis and José A. R. Fonollosa. 2012. “Dealing with input noise in statistical machine translation.” *COLING*. Mumbai, India.

Foster, George and Roland Kuhn. 2009. “Stabilizing minimum error rate training.” *WMT*. Athens, Greece.

Gerlach, Johanna, Victoria Porro, Pierrette Bouillon, and Sabine Lehmann. 2013. “La préédition avec des règles peu coûteuses, utile pour la TA statistique?” *TALN-RECITAL*. Sables d’Olonne, France.

Kneser, Reinhard and Hermann Ney. 1995. “Improved backing-off for m-gram language modeling.” *ICASSP*. Detroit, MI, USA.

Koehn, Philipp. 2005. “Europarl: A parallel corpus for statistical machine translation.” *MT Summit X*. Phuket, Thailand.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, et al. 2007. “Moses: Open source toolkit for statistical machine translation.” *ACL Demonstration Session*. Prague, Czech Republic.

Och, Franz Josef and Hermann Ney. 2000. “Improved statistical alignment models.” *ACL*. Hong Kong.

Rabiner, Lawrence R. 1989. “A tutorial on hidden markov models and selected applications in speech recognition.” *Proceedings of the IEEE*, 257–286.

Rayner, Manny, Pierrette Bouillon, and Barry Haddow. 2012. “Using source-language transformations to address register mismatches in SMT.” *AMTA*. San Diego, CA, USA.

Stolcke, Andreas. 2002. “SRILM - an extensible language modeling toolkit.” *ICSLP*. Denver, CO, USA.

Edinburgh's Machine Translation Systems for European Language Pairs

Nadir Durrani, Barry Haddow, Kenneth Heafield, and Philipp Koehn
School of Informatics
University of Edinburgh
Scotland, United Kingdom
{dnadir,bhaddow,kheafiel,pkoehn}@inf.ed.ac.uk

Abstract

We validated various novel and recently proposed methods for statistical machine translation on 10 language pairs, using large data resources. We saw gains from optimizing parameters, training with sparse features, the operation sequence model, and domain adaptation techniques. We also report on utilizing a huge language model trained on 126 billion tokens.

The annual machine translation evaluation campaign for European languages organized around the ACL Workshop on Statistical Machine Translation offers the opportunity to test recent advancements in machine translation in large data condition across several diverse language pairs.

Building on our own developments and external contributions to the Moses open source toolkit, we carried out extensive experiments that, by early indications, led to a strong showing in the evaluation campaign.

We would like to stress especially two contributions: the use of the new operation sequence model (Section 3) within Moses, and — in a separate unconstraint track submission — the use of a huge language model trained on 126 billion tokens with a new training tool (Section 4).

1 Initial System Development

We start with systems (Haddow and Koehn, 2012) that we developed for the 2012 Workshop on Statistical Machine Translation (Callison-Burch et al., 2012). The notable features of these systems are:

- Moses phrase-based models with mostly default settings
- training on all available parallel data, including the large UN parallel data, the French-English 10^9 parallel data and the LDC Gigaword data
- very large tuning set consisting of the test sets from 2008-2010, with a total of 7,567 sentences per language
- German–English with syntactic pre-reordering (Collins et al., 2005), compound splitting (Koehn and Knight, 2003) and use of factored representation for a POS target sequence model (Koehn and Hoang, 2007)
- English–German with morphological target sequence model

Note that while our final 2012 systems included subsampling of training data with modified Moore-Lewis filtering (Axelrod et al., 2011), we did not use such filtering at the starting point of our development. We will report on such filtering in Section 2.

Moreover, our system development initially used the WMT 2012 data condition, since it took place throughout 2012, and we switched to WMT 2013 training data at a later stage. In this section, we report cased BLEU scores (Papineni et al., 2001) on newstest2011.

1.1 Factored Backoff (German–English)

We have consistently used factored models in past WMT systems for the German–English language pairs to include POS and morphological target sequence models. But we did not use the factored decomposition of translation options into multiple mapping steps, since this usually lead to much slower systems with usually worse results.

A good place, however, for factored decomposition is the handling of rare and unknown source words which have more frequent morphological variants (Koehn and Haddow, 2012a). Here, we used only factored backoff for unknown words, giving gains in BLEU of +.12 for German–English.

1.2 Tuning with k-best MIRA

In preparation for training with sparse features, we moved away from MERT which is known to fall

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 114–121,
Sofia, Bulgaria, August 8-9, 2013

apart with many more than a couple of dozen features. Instead, we used k-best MIRA (Cherry and Foster, 2012). For the different language pairs, we saw improvements in BLEU of -.05 to +.39, with an average of +.09. There was only a minimal change in the length ratio (Table 1)

	MERT	k-best MIRA	Δ
de-en	22.11 (1.010)	22.10 (1.008)	–.01 (+.002)
fr-en	30.00 (1.023)	30.11 (1.026)	+.11 (±.003)
es-en	30.42 (1.021)	30.63 (1.020)	+.21 (–.001)
cs-en	25.54 (1.022)	25.49 (1.024)	–.05 (±.002)
en-de	16.08 (0.995)	16.04 (1.001)	–.04 (±.006)
en-fr	29.26 (0.980)	29.65 (0.982)	+.39 (±.002)
en-es	31.92 (0.985)	31.95 (0.985)	+.03 (±.000)
en-cs	17.38 (0.967)	17.42 (0.974)	+.04 (±.007)
avg	–	–	+.09

Table 1: Tuning with k-best MIRA instead of MERT (cased BLEU scores with length ratio)

1.3 Translation Table Smoothing with Kneser-Ney Discounting

Previously, we smoothed counts for the phrasal conditional probability distributions in the translation model with Good Turing discounting. We explored the use of Kneser-Ney discounting, but results are mixed (no difference on average, see Table 2), so we did not pursue this further.

	Good Turing	Kneser Ney	Δ
de-en	22.10	22.15	+.05
fr-en	30.11	30.13	+.02
es-en	30.63	30.64	+.01
cs-en	25.49	25.56	+.07
en-de	16.04	15.93	–.11
en-fr	29.65	29.75	+.10
en-es	31.95	31.98	+.03
en-cs	17.42	17.26	–.16
avg	–	–	±.00

Table 2: Translation model smoothing with Kneser-Ney

1.4 Sparse Features

A significant extension of the Moses system over the last couple of years was the support for large numbers of sparse features. This year, we tested this capability on our big WMT systems. First, we used features proposed by Chiang et al. (2009):

- phrase pair count bin features (bins 1, 2, 3, 4–5, 6–9, 10+)
- target word insertion features
- source word deletion features
- word translation features
- phrase length feature (source, target, both)

The lexical features were restricted to the 50 most frequent words. All these features together only gave minor improvements (Table 3).

	baseline	sparse	Δ
de-en	22.10	22.02	–.08
fr-en	30.11	30.24	+.13
es-en	30.63	30.61	–.02
cs-en	25.49	25.49	±.00
en-de	16.04	15.93	–.09
en-fr	29.65	29.81	+.16
en-es	31.95	32.02	+.07
en-cs	17.42	17.28	–.14
avg	–	–	+.04

Table 3: Sparse features

We also explored domain features in the sparse feature framework, in three different variations. Assume that we have three domains, and a phrase pair occurs in domain A 15 times, in domain B 5 times, and in domain C never.

We compute three types of domain features:

- binary indicator, if phrase-pairs occurs in domain (example: $\text{ind}_A = 1$, $\text{ind}_B = 1$, $\text{ind}_C = 0$)
- ratio how frequent the phrase pairs occurs in domain (example: $\text{ratio}_A = \frac{15}{15+5} = .75$, $\text{ratio}_B = \frac{5}{15+5} = .25$, $\text{ratio}_C = 0$)
- subset of domains in which phrase pair occurs (example: $\text{subset}_{AB} = 1$, other subsets 0)

We tested all three feature types, and found the biggest gain with the domain indicator feature (+.11, Table 4). Note that we define as domain the different corpora (Europarl, etc.). The number of domains ranges from 2 to 9 (see column #d).[1]

	#d	base.	indicator	ratio	subset
de-en	2	22.10	22.14 +.04	22.07 –.03	22.12 +.02
fr-en	4	30.11	30.34 +.23	30.29 +.18	30.15 +.04
es-en	3	30.63	30.88 +.25	30.64 +.01	30.82 +.19
cs-en	9	25.49	25.58 +.09	25.58 +.09	25.46 –.03
en-de	2	16.12[2]	16.14 +.02	15.96 –.16	16.01 –.11
en-fr	4	29.65	29.75 +.10	29.71 +.05	29.70 +.05
en-es	3	31.95	32.06 +.11	32.13 +.18	32.02 +.07
en-cs	9	17.42	17.45 +.03	17.35 –.07	17.44 +.02
avg.	-	–	+.11	+.03	+.03

Table 4: Sparse domain features

When combining the domain features and the other sparse features, we see roughly additive gains (Table 5). We use the domain indicator feature and the other sparse features in subsequent experiments.

[1] In the final experiments on the 2013 data condition, one domain (*commoncrawl*) was added for all language pairs.

	baseline	indicator	ratio	subset
de-en	22.10	22.18 +.08	22.10 ±.00	22.16 +.06
fr-en	30.11	30.41 +.30	30.49 +.38	30.36 +.25
es-en	30.63	30.75 +.12	30.56 –.07	30.85 +.22
cs-en	25.49	25.56 +.07	25.63 +.14	25.43 –.06
en-de	16.12	15.95 –.17	15.96 –.16	16.05 –.07
en-fr	29.65	29.96 +.31	29.88 +.23	29.92 +.27
en-es	31.95	32.12 +.17	32.16 +.21	32.08 +.23
en-cs	17.42	17.38 –.04	17.35 –.07	17.40 –.02
avg.	–	+.11	+.09	+.11

Table 5: Combining domain and other sparse features

1.5 Tuning Settings

Given the opportunity to explore the parameter tuning of models with sparse features across many language pairs, we investigated a number of settings. We expect tuning to work better with more iterations, longer n-best lists and bigger cube pruning pop limits. Our baseline settings are 10 iterations with 100-best lists (accumulating) and a pop limit of 1000 for tuning and 5000 for testing.

	base	25 it.	25it+1k-best	25it+pop5k
de-en	22.18	22.16 –.02	22.14 –.04	22.17 –.01
fr-en	30.41	30.40 –.01	30.44 +.03	30.49 +.08
es-en	30.75	30.91 +.16	30.86 +.11	30.81 +.06
cs-en	25.56	25.60 +.04	25.64 +.08	25.56 ±.00
en-de	15.96	15.99 +.03	16.05 +.09	15.96 ±.00
en-fr	29.96	29.90 –.06	29.95 –.01	29.92 –.04
en-es	32.12	32.17 +.05	32.11 –.01	32.19 +.07
en-cs	17.38	17.43 +.05	17.50 +.12	17.38 ±.00
avg	–	+.03	+.05	+.02

Table 6: Tuning settings (number of iterations, size of n-best list, and cube pruning pop limit)

Results support running tuning for 25 iterations but we see no gains for 5000 pops. There is evidence that an n-best list size of 1000 is better in tuning but we did not adopt this since these large lists take up a lot of disk space and slow down the MIRA optimization step (Table 6).

1.6 Smaller Phrases

Given the very large corpus sizes (up to a billion words of parallel data for French–English), the size of translation model and lexicalized reordering model becomes a challenge. Hence, we want to examine if restriction to smaller phrases is feasible without loss in translation quality. Results in Table 7 suggest that a maximum phrase length of 5 gives almost identical results, and only with a phrase length limit of 4 significant losses occur. We adopted the limit of 5.

	max 7	max 6	max 5	max 4
de-en	22.16	22.03 –.13	22.05 –.11	22.17 +.01
fr-en	30.40	30.30 –.10	30.39 –.01	30.23 –.17
es-en	30.91	30.80 –.09	30.86 –.05	30.81 –.10
cs-en	25.60	25.55 –.05	25.53 –.07	25.48 –.12
en-de	15.99	15.94 –.05	15.97 –.02	16.03 +.04
en-fr	29.90	29.97 +.07	29.89 –.01	29.77 –.13
en-es	32.17	32.13 –.04	32.27 +.10	31.93 –.24
en-cs	17.43	17.46 +.03	17.41 –.02	17.41 –.02
avg	–	–.05	–.03	–.09

Table 7: Maximum phrase length, reduced from baseline

1.7 Unpruned Language Models

Previously, we trained 5-gram language models using the default settings of the SRILM toolkit in terms of singleton pruning. Thus, training throws out all singletons n-grams of order 3 and higher. We explored whether unpruned language models could give better performance, even if we are only able to train 4-gram models due to memory constraints. At the time, we were not able to build unpruned 4-gram language models for English, but for the other language pairs we did see improvements of -.07 to +.13 (Table 8). We adopted such models for these language pairs.

	5g pruned	4g unpruned	Δ
en-fr	29.89	29.83	–.07
en-es	32.27	32.34	+.07
en-cs	17.41	17.54	+.13

Table 8: Language models without singleton pruning

1.8 Translations per Input Phrase

Finally, we explored one more parameter: the limit on how many translation options are considered per input phrase. The default for this setting is 20. However, our experiments (Table 9) show that we can get better results with a translation table limit of 100, so we adopted this.

	ttl 20	ttl 30	ttl 50	ttl 100
de-en	21.05	+.06	+.09	+.01
fr-en	30.39	–.02	+.05	+.07
es-en	30.86	±.00	–.03	–.07
cs-en	25.53	+.24	+.13	+.20
en-de	15.97	+.03	+.07	+.11
en-fr	29.83	+.14	+.19	+.13
en-es	32.34	+.08	+.10	+.07
en-cs	17.54	–.05	–.02	+.01
avg	–	+.06	+.07	+.07

Table 9: Maximal number translations per input phrase

1.9 Other Experiments

We explored a number of other settings and features, but did not observe any gains.

- Using HMM alignment instead of IBM Model 4 leads to losses of –.01 to –.27.
- An earlier check of modified Moore–Lewis filtering (see also below in Section 3) gave very inconsistent results.
- Filtering the phrase table with significance filtering (Johnson et al., 2007) leads to losses of –.19 to –.63.
- Throwing out phrase pairs with direct translation probability $\phi(\bar{e}|\bar{f})$ of less than 10^{-5} has almost no effect.
- Double-checking the contribution of the sparse lexical features in the final setup, we observe an average losses of –.07 when dropping these features.
- For the German–English language pairs we saw some benefits to using sparse lexical features over POS tags instead of words, so we used this in the final system.

1.10 Summary

We adopted a number of changes that improved our baseline system by an average of +.30, see Table 10 for a breakdown.

avg.	method
+.01	factored backoff
+.09	kbest MIRA
+.11	sparse features and domain indicator
+.03	tuning with 25 iterations
–.03	maximum phrase length 5
+.02	unpruned 4-gram LM
+.07	translation table limit 100
+.30	total

Table 10: Summary of impact of changes

Minor improvements that we did not adopt was avoiding reducing maximum phrase length to 5 (average +.03) and tuning with 1000-best lists (+.02).

The improvements differed significantly by language pair, as detailed in Table 11, with the biggest gains for English–French (+.70), no gain for English–German and no gain for English–German.

1.11 New Data

The final experiment of the initial system development phase was to train the systems on the new data, adding newstest2011 to the tuning set (now 10,068 sentences). Table 12 reports the gains on newstest2012 due to added data, indicating very clearly that valuable new data resources became available this year.

	baseline	improved	Δ
de-en	21.99	22.09	+.10
fr-en	30.00	30.46	+.46
es-en	30.42	30.79	+.37
cs-en	25.54	25.73	+.19
en-de	16.08	16.08	±.00
en-fr	29.26	29.96	+.70
en-es	31.92	32.41	+.49
en-cs	17.38	17.55	+.17

Table 11: Overall improvements per language pair

	WMT 2012	WMT 2013	Δ
de-en	23.11	24.01	+0.90
fr-en	29.25	30.77	+1.52
es-en	32.80	33.99	+1.19
cs-en	22.53	22.86	+0.33
ru-en	–	31.67	–
en-de	16.78	17.95	+1.17
en-fr	27.92	28.76	+0.84
en-es	33.41	34.00	+0.59
en-cs	15.51	15.78	+0.27
en-ru	–	23.78	–

Table 12: Training with new data (newstest2012 scores)

2 Domain Adaptation Techniques

We explored two additional domain adaptation techniques: phrase table interpolation and modified Moore-Lewis filtering.

2.1 Phrase Table Interpolation

We experimented with phrase-table interpolation using perplexity minimisation (Foster et al., 2010; Sennrich, 2012). In particular, we used the implementation released with Sennrich (2012) and available in Moses, comparing both the **naive** and **modified** interpolation methods from that paper. For each language pair, we took the alignments created from all the data concatenated, built separate phrase tables from each of the individual corpora, and interpolated using each method. The results are shown in Table 13

	baseline	naive	modified
fr-en	**30.77**	30.63 –.14	–
es-en*	33.98	33.83 –.15	**34.03** +.05
cs-en*	**23.19**	22.77 –.42	23.03 –.17
ru-en	**31.67**	31.42 –.25	31.59 –.08
en-fr	28.76	**28.88** +.12	–
en-es	34.00	34.07 +.07	**34.31** +.31
en-cs	15.78	**15.88** +.10	15.87 +.09
en-ru	23.78	**23.84** +.06	23.68 –.10

Table 13: Comparison of phrase-table interpolation (two methods) with baseline (on newstest2012). The baselines are as Table 12 except for the starred rows where tuning with PRO was found to be better. The modified interpolation was not possible in fr↔en as it uses to much RAM.

The results from the phrase-table interpolation are quite mixed, and we only used the technique

for the final system in en-es. An interpolation based on PRO has recently been shown (Haddow, 2013) to improve on perplexity minimisation is some cases, but the current implementation of this method is limited to 2 phrase-tables, so we did not use it in this evaluation.

2.2 Modified Moore-Lewis Filtering

In last year's evaluation (Koehn and Haddow, 2012b) we had some success with modified Moore-Lewis filtering (Moore and Lewis, 2010; Axelrod et al., 2011) of the training data. This year we conducted experiments in most of the language pairs using MML filtering, and also experimented using *instance weighting* (Mansour and Ney, 2012) using the (exponential of) the MML weights. The results are show in Table 14

	base line	MML 20%	Inst. Wt	Inst. Wt (scale)
fr-en	**30.77**	–	–	–
es-en*	33.98	**34.26** +.28	33.85 –.13	33.98 ±.00
cs-en*	**23.19**	22.62 –.57	23.17 –.02	23.13 –.06
ru-en	**31.67**	31.58 –.09	31.57 –.10	31.62 –.05
en-fr	28.67	28.74 +.07	**28.81** +.17	28.63 –.04
en-es	34.00	34.07 +.07	**34.27** +.27	34.03 +.03
en-cs	15.78	15.37 –.41	15.87 +.09	**15.89** +.11
en-ru	23.78	22.90 –.88	**23.82** +.05	23.72 –.06

Table 14: Comparison of MML filtering and weighting with baseline. The MML uses monolingual news as in-domain, and selects from all training data after alignment.The weighting uses the MML weights, optionally downscaled by 10, then exponentiated. Baselines are as Table 13.

As with phrase-table interpolation, MML filtering and weighting shows a very mixed picture, and not the consistent improvements these techniques offer on IWSLT data. In the final systems, we used MML filtering only for es-en.

3 Operation Sequence Model (OSM)

We enhanced the phrase segmentation and reordering mechanism by integrating OSM: an operation sequence N-gram-based translation and reordering model (Durrani et al., 2011) into the Moses phrase-based decoder. The model is based on minimal translation units (MTUs) and Markov chains over sequences of operations. An operation can be (a) to jointly generate a bi-language MTU, composed from source and target words, or (b) to perform reordering by inserting gaps and doing jumps.

Model: Given a bilingual sentence pair $< F, E >$ and its alignment A, we transform it to

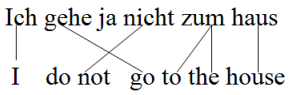


Figure 1: Bilingual Sentence with Alignments

sequence of operations $(o_1, o_2, \ldots, o_J)$ and learn a Markov model over this sequence as:

$$p_{osm}(F, E, A) = p(o_1^J) = \prod_{j=1}^{J} p(o_j | o_{j-n+1}, ..., o_{j-1})$$

By coupling reordering with lexical generation, each (translation or reordering) decision conditions on $n-1$ previous (translation and reordering) decisions spanning across phrasal boundaries thus overcoming the problematic phrasal independence assumption in the phrase-based model. In the OSM model, the reordering decisions influence lexical selection and vice versa. Lexical generation is strongly coupled with reordering thus improving the overall reordering mechanism.

We used the modified version of the OSM model (Durrani et al., 2013b) that additionally handles discontinuous and unaligned target MTUs[3]. We borrow 4 count-based supportive features, the *Gap*, *Open Gap*, *Gap-width* and *Deletion* penalties from Durrani et al. (2011).

Training: During training, each bilingual sentence pair is deterministically converted to a unique sequence of operations. Please refer to Durrani et al. (2011) for a list of operations and the conversion algorithm and see Figure 1 and Table 15 for a sample bilingual sentence pair and its step-wise conversion into a sequence of operation. A 9-gram Kneser-Ney smoothed operation sequence model is trained with SRILM.

Search: Although the OSM model is based on minimal units, phrase-based search on top of OSM model was found to be superior to the MTU-based decoding in Durrani et al. (2013a). Following this framework allows us to use OSM model in tandem with phrase-based models. We integrated the generative story of the OSM model into the hypothesis extension of the phrase-based Moses decoder. Please refer to (Durrani et al., 2013b) for details.

Results: Table 16 shows case-sensitive BLEU scores on newstest2012 and newstest2013 for fi-

[3]In the original OSM model these are removed from the alignments through a post-processing heuristic which hurts in some language pairs. See Durrani et al. (2013b) for detailed experiments.

Operation Sequence	Generation
Generate(Ich, I)	Ich ↓ I
Generate Target Only (do)	Ich ↓ I do
Insert Gap Generate (nicht, not)	Ich ☐ nicht ↓ I do not
Jump Back (1) Generate (gehe, go)	Ich gehe ↓ nicht I do not go
Generate Source Only (ja)	Ich gehe ja ↓ nicht I do not go
Jump Forward	Ich gehe ja nicht ↓ I do not go
Generate (zum, to the)	... gehe ja nicht zum ↓ ... not go to the
Generate (haus, house)	... ja nicht zum haus ↓ ... go to the house

Table 15: Step-wise Generation of Figure 1

LP	Baseline		+OSM	
newstest	2012	2013	2012	2013
de-en	23.85	26.54	24.11 +.26	26.83 +.29
fr-en	30.77	31.09	30.96 +.19	31.46 +.37
es-en	34.02	30.04	34.51 +.49	30.94 +.90
cs-en	22.70	25.70	23.03 +.33	25.79 +.09
ru-en	31.87	24.00	32.33 +.46	24.33 +.33
en-de	17.95	20.06	18.02 +.07	20.26 +.20
en-fr	28.76	30.03	29.36 +.60	30.39 +.36
en-es	33.87	29.66	34.44 +.57	30.10 +.44
en-cs	15.81	18.35	16.16 +.35	18.62 +.27
en-ru	23.75	18.44	24.05 +.30	18.84 +.40

Table 16: Results using the OSM Feature

nal systems from Section 1 and these systems augmented with the operation sequence model. The model gives gains for all language pairs (BLEU +.09 to +.90, average +.37, on newstest2013).

4 Huge Language Models

To overcome the memory limitations of SRILM, we implemented modified Kneser-Ney (Kneser and Ney, 1995; Chen and Goodman, 1998) smoothing from scratch using disk-based streaming algorithms. This open-source[4] tool is described fully by Heafield et al. (2013). We used it to estimate an unpruned 5–gram language model on web pages from ClueWeb09.[5] The corpus was preprocessed by removing spam (Cormack et al., 2011), selecting English documents, splitting sentences, deduplicating, tokenizing, and truecasing. Estimation on the remaining 126 billion tokens took 2.8 days on a single machine with 140 GB RAM (of which 123 GB was used at peak) and six hard drives in a RAID5 configuration. Statistics about the resulting model are shown in Table 17.

[4] http://kheafield.com/code/

[5] http://lemurproject.org/clueweb09/

1	2	3	4	5
393m	3,775m	17,629m	39,919m	59,794m

Table 17: Counts of unique n-grams (m for millions) for the 5 orders in the unconstrained language model

The large language model was then quantized to 10 bits and compressed to 643 GB with KenLM (Heafield, 2011), loaded onto a machine with 1 TB RAM, and used as an additional feature in unconstrained French–English, Spanish–English, and Czech–English submissions. This additional language model is the only difference between our final constrained and unconstrained submissions; no additional parallel data was used. Results are shown in Table 18. Improvement from large language models is not a new result (Brants et al., 2007); the primary contribution is estimating on a single machine.

	Constrained	Unconstrained	Δ
fr-en	31.46	32.24	+.78
es-en	30.59	31.37	+.78
cs-en	27.38	28.16	+.78
ru-en	24.33	25.14	+.81

Table 18: Gain on newstest2013 from the unconstrained language model. Our time on shared machines with 1 TB is limited so Russian–English was run after the deadline and German–English was not ready in time.

5 Summary

Table 19 breaks down the gains over the final system from Section 1 from using the operation sequence models (OSM), modified Moore-Lewis filtering (MML), fixing a bug with the sparse lexical features (Sparse-Lex Bugfix), and instance weighting (Instance Wt.), translation model combination (TM-Combine), and use of the huge language model (ClueWeb09 LM).

Acknowledgments

Thanks to Miles Osborne for preprocessing the ClueWeb09 corpus. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 287658 (EU BRIDGE) and grant agreement 288487(MosesCore).This work made use of the resources provided by the Edinburgh Compute and Data Facility[6]. The ECDF is partially supported by the eDIKT initiative[7]. This work also used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, Stampede was used under allocation TG-CCR110017.

[6] http://www.ecdf.ed.ac.uk/

[7] http://www.edikt.org.uk/

	System	2012	2013
	Spanish-English		
1.	Baseline	34.02	30.04
2.	1+OSM	34.51 +.49	30.94 +.90
3.	1+MML (20%)	34.38 +.36	30.38 +.34
4.	1+Sparse-Lex Bugfix	34.17 +.15	30.33 +.29
5.	1+2+3: OSM+MML	34.65 +.63	30.51 +.47
6.	**1+2+3+4**	34.68 +.66	30.59 +.55
7.	**6+ClueWeb09 LM**		31.37 +1.33
	English-Spanish		
1.	Baseline	33.87	29.66
2.	1+OSM	34.44 +.57	30.10 +.44
3.	1+TM-Combine	34.31 +44	29.76 +.10
4.	1+Instance Wt.	34.27 +.40	29.63 –.03
5.	1+Sparse-Lex Bugfix	34.20 +.33	29.86 +.20
6.	1+2+3: OSM+TM-Cmb.	34.63 +.76	30.21 +.55
7.	1+2+4: OSM+Inst. Wt.	34.58 +.71	30.11 +.45
8.	**1+2+3+5**	34.78 +.91	30.43 +.77
	Czech-English		
1.	Baseline	22.70	25.70
2.	1+OSM	23.03 +.33	25.79 +.09
3.	1+with PRO	23.19 +.49	26.08 +.38
4.	1+Sparse-Lex Bugfix	22.86 +.16	25.74 +.04
5.	**1+OSM+PRO**	23.42 +.72	26.23 +.53
6.	1+2+3+4	23.16 +.46	25.94 +.24
7.	**5+ClueWeb09 LM**		27.06 +.36
	English-Czech		
1.	Baseline	15.85	18.35
2.	**1+OSM**	16.16 +.31	18.62 +.27
	French-English		
1.	Baseline	30.77	31.09
2.	**1+OSM**	30.96 +.19	31.46 +.37
3.	**2+ClueWeb09 LM**		32.24 +1.15
	English-French		
1.	Baseline	28.76	30.03
2.	1+OSM	29.36 +.60	30.39 +.36
3.	1+Sparse-Lex Bugfix	28.97 +.21	30.08 +.05
4.	**1+2+3**	29.37 +.61	30.58 +.55
	German-English		
1.	**Baseline**	23.85	26.54
2.	1+OSM	24.11 +.26	26.83 +.29
	English-German		
1.	**Baseline**	17.95	20.06
2.	1+OSM	18.02 +.07	20.26 +.20
	Russian-English		
1.	Baseline	31.87	24.00
2.	**1+OSM**	32.33 +.46	24.33 +.33
	English-Russian		
1.	Baseline	23.75	18.44
2.	**1+OSM**	24.05 +.40	18.84 +.40

Table 19: Summary of methods with BLEU scores on newstest2012 and newstest2013. Bold systems were submitted, with the ClueWeb09 LM systems submitted in the unconstraint track. The German–English and English–German OSM systems did not complete in time for the official submission.

References

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–48, Montreal, Canada. Association for Computational Linguistics.

Chen, S. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado. 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.

Cormack, G. V., Smucker, M. D., and Clarke, C. L. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465.

Durrani, N., Fraser, A., and Schmid, H. (2013a). Model With Minimal Translation Units, But Decode With Phrases. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA. Association for Computational Linguistics.

Durrani, N., Fraser, A., Schmid, H., Hoang, H., and Koehn, P. (2013b). Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria. Association for Computational Linguistics.

Durrani, N., Schmid, H., and Fraser, A. (2011). A Joint Sequence Translation Model with Integrated Reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054, Portland, Oregon, USA.

Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA. Association for Computational Linguistics.

Haddow, B. (2013). Applying pairwise ranked optimisation to improve the interpolation of translation models. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 342–347, Atlanta, Georgia. Association for Computational Linguistics.

Haddow, B. and Koehn, P. (2012). Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 175–185, Montreal, Canada. Association for Computational Linguistics.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria.

Johnson, H., Martin, J., Foster, G., and Kuhn, R. (2007). Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184.

Koehn, P. and Haddow, B. (2012a). Interpolated backoff for factored translation models. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*.

Koehn, P. and Haddow, B. (2012b). Towards Effective Use of Training Data in Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 317–321, Montréal, Canada. Association for Computational Linguistics.

Koehn, P. and Hoang, H. (2007). Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic. Association for Computational Linguistics.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*.

Mansour, S. and Ney, H. (2012). A Simple and Effective Weighted Phrase Extraction for Machine Translation Adaptation. In *Proceedings of IWSLT*.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176(W0109-022), IBM Research Report.

Sennrich, R. (2012). Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France. Association for Computational Linguistics.

Applying Pairwise Ranked Optimisation to Improve the Interpolation of Translation Models

Barry Haddow
University of Edinburgh
Scotland
bhaddow@inf.ed.ac.uk

Abstract

In Statistical Machine Translation we often have to combine different sources of parallel training data to build a good system. One way of doing this is to build separate translation models from each data set and linearly interpolate them, and to date the main method for optimising the interpolation weights is to minimise the model perplexity on a heldout set. In this work, rather than optimising for this indirect measure, we directly optimise for BLEU on the tuning set and show improvements in average performance over two data sets and 8 language pairs.

1 Introduction

Statistical Machine Translation (SMT) requires large quantities of parallel training data in order to produce high quality translation systems. This training data, however, is often scarce and must be drawn from whatever sources are available. If these data sources differ systematically from each other, and/or from the test data, then the problem of combining these disparate data sets to create the best possible translation system is known as *domain adaptation*.

One approach to domain adaptation is to build separate models for each training domain, then weight them to create a system tuned to the test domain. In SMT, a successful approach to building domain specific language models is to build one from each corpus, then linearly interpolate them, choosing weights that minimise the perplexity on a suitable heldout set of in-domain data. This method has been applied by many authors (e.g. (Koehn and Schroeder, 2007)), and is implemented in popular language modelling tools like IRSTLM (Federico et al., 2008) and SRILM (Stolcke, 2002).

Similar interpolation techniques have been developed for translation model interpolation (Foster et al., 2010; Sennrich, 2012) for phrase-based systems but have not been as widely adopted, perhaps because the efficacy of the methods is not as clear-cut. In this previous work, the authors used standard phrase extraction heuristics to extract phrases from a heldout set of parallel sentences, then tuned the translation model (i.e. the phrase table) interpolation weights to minimise the perplexity of the interpolated model on this set of extracted phrases.

In this paper, we try to improve on this perplexity optimisation of phrase table interpolation weights by addressing two of its shortcomings. The first problem is that the perplexity is not well defined because of the differing coverage of the phrase tables, and their partial coverage of the phrases extracted from the heldout set. Secondly, perplexity may not correlate with the performance of the final SMT system.

So, instead of optimising the interpolation weights for the indirect goal of translation model perplexity, we optimise them directly for translation performance. We do this by incorporating these weights into SMT tuning using a modified version of Pairwise Ranked Optimisation (PRO) (Hopkins and May, 2011).

In experiments on two different domain adaptation problems and 8 language pairs, we show that our method achieves comparable or improved performance, when compared to the perplexity minimisation method. This is an encouraging result as it

shows that PRO can be adapted to optimise translation parameters other than those in the standard linear model.

2 Optimising Phrase Table Interpolation Weights

2.1 Previous Approaches

In the work of Foster and Kuhn (2007), linear interpolation weights were derived from different measures of distance between the training corpora, but this was not found to be successful. Optimising the weights to minimise perplexity, as described in the introduction, was found by later authors to be more useful (Foster et al., 2010; Sennrich, 2012), generally showing small improvements over the default approach of concatenating all training data.

An alternative approach is to use log-linear interpolation, so that the interpolation weights can be easily optimised in tuning (Koehn and Schroeder, 2007; Bertoldi and Federico, 2009; Banerjee et al., 2011). However, this effectively multiplies the probabilities across phrase tables, which does not seem appropriate, especially for phrases absent from 1 table.

2.2 Tuning SMT Systems

The standard SMT model scores translation hypotheses as a linear combination of features. The model score of a hypothesis e is then defined to be $\mathbf{w} \cdot \mathbf{h}(e, f, a)$ where $\mathbf{w}$ is a weight vector, and $\mathbf{h}(e, f, a)$ a vector of feature functions defined over source sentences (f), hypotheses, and their alignments (a). The weights are normally optimised (*tuned*) to maximise BLEU on a heldout set (the *tuning* set).

The most popular algorithm for this weight optimisation is the line-search based MERT (Och, 2003), but recently other algorithms that support more features, such as PRO (Hopkins and May, 2011) or MIRA-based algorithms (Watanabe et al., 2007; Chiang et al., 2008; Cherry and Foster, 2012), have been introduced. All these algorithms assume that the model score is a linear function of the parameters $\mathbf{w}$. However since the phrase table probabilities enter the score function in log form, if these probabilities are a linear interpolation, then the model score is not a linear function of the interpolation weights. We will show that PRO can be used to simultaneously optimise such non-linear parameters.

2.3 Pairwise Ranked Optimisation

PRO is a *batch* tuning algorithm in the sense that there is an outer loop which repeatedly decodes a small (1000-2000 sentence) tuning set and passes the n-best lists from this tuning set to the core algorithm (also known as the *inner loop*). The core algorithm samples pairs of hypotheses from the n-best lists (according to a specific procedure), and uses these samples to optimise the weight vector $\mathbf{w}$.

The core algorithm in PRO will now be explained in more detail. Suppose that the N sampled hypothesis pairs (x_i^α, x_i^β) are indexed by i and have corresponding feature vectors pairs $(\mathbf{h}_i^\alpha, \mathbf{h}_i^\beta)$. If the gain of a given hypothesis (we use smoothed sentence BLEU) is given by the function $g(x)$, then we define y_i by

$$y_i \equiv \text{sgn}(g(x_i^\alpha) - g(x_i^\beta)) \quad (1)$$

For weights $\mathbf{w}$, and hypothesis pair (x_i^α, x_i^β), the (model) score difference $\Delta s_i^\mathbf{w}$ is given by:

$$\Delta s_i^\mathbf{w} \equiv s^\mathbf{w}(x_i^\alpha) - s^\mathbf{w}(x_i^\beta) \equiv \mathbf{w} \cdot \left(\mathbf{h}_i^\alpha - \mathbf{h}_i^\beta\right) \quad (2)$$

Then the core PRO algorithm updates the weight vector to $\mathbf{w}^*$ by solving the following optimisation problem:

$$\mathbf{w}^* = \arg\max_\mathbf{w} \sum_{i=1}^{N} \log\left(\sigma\left(y_i \Delta s_i^\mathbf{w}\right)\right) \quad (3)$$

where $\sigma(x)$ is the standard sigmoid function. The derivative of the function can be computed easily, and the optimisation problem can be solved with standard numerical optimisation algorithms such as L-BFGS (Byrd et al., 1995). PRO is normally implemented by converting each sample to a training example for a 2 class maximum entropy classifier, with the feature values set to $\Delta\mathbf{h}_i$ and the responses set to the y_i, whereupon the log-likelihood is the objective given in Equation (3). As in maximum entropy modeling, it is usual to add a Gaussian prior to the objective (3) in PRO training.

2.4 Extending PRO for Mixture Models

We now show how to apply the PRO tuning algorithm of the previous subsection to simultaneously

optimise the weights of the translation system, and the interpolation weights.

In the standard phrase-based model, some of the features are derived from logs of phrase translation probabilities. If the phrase table is actually a linear interpolation of two (or more) phrase tables, then we can consider these features as also being functions of the interpolation weights. The interpolation weights then enter the score differences $\{\Delta s_i^{\mathbf{w}}\}$ via the phrase features, and we can jointly optimise the objective in Equation (3) for translation model weights and interpolation weights.

To make this more concrete, suppose that the feature vector consists of m phrase table features and $n - m$ other features[1]

$$\mathbf{h} \equiv (\log(p^1), \ldots, \log(p^m), h^{m+1}, \ldots h^n) \quad (4)$$

where each p^j is an interpolation of two probability distributions p_A^j and p_B^j. So, $p^j \equiv \lambda^j p_A^j + (1-\lambda^j) p_B^j$ with $0 \leq \lambda^j \leq 1$. Defining $\boldsymbol{\lambda} \equiv (\lambda^1 \ldots \lambda^m)$, the optimisation problem is then:

$$(\mathbf{w}^*, \boldsymbol{\lambda}^*) = \arg\max_{(\mathbf{w},\boldsymbol{\lambda})} \sum_{i=1}^{N} \log\left(\sigma\left(y_i \Delta s_i^{(\mathbf{w},\boldsymbol{\lambda})}\right)\right) \quad (5)$$

where the sum is over the sampled hypothesis pairs and the Δ indicates the difference between the model scores of the two hypotheses in the pair, as before. The model score $s_i^{(\mathbf{w},\boldsymbol{\lambda})}$ is given by

$$\sum_{j=1}^{m} \left(w^j \cdot \log\left(\lambda^j p_{Ai}^j + (1-\lambda^j) p_{Bi}^j\right)\right) + \sum_{j=m+1}^{n} w^j h_i^j \quad (6)$$

where $\mathbf{w} \equiv (w^i \ldots w^n)$. A Gaussian regularisation term is added to the objective, as it was for PRO. By replacing the core algorithm of PRO with the optimisation above, the interpolation weights can be trained simultaneously with the other model weights.

Actually, the above explanation contains a simplification, in that it shows the phrase features interpolated at sentence level. In reality the phrase features are interpolated at the phrase level, then combined to give the sentence level feature value. This makes the definition of the objective more complex than that shown above, but still optimisable using bounded L-BFGS.

[1] Since the phrase penalty feature is a constant across phrase pairs it is not interpolated, and so is classed with the the "other" features. The lexical scores, although not actually probabilities, are interpolated.

3 Experiments

3.1 Corpus and Baselines

We ran experiments with data from the WMT shared tasks (Callison-Burch et al., 2007; Callison-Burch et al., 2012), as well as OpenSubtitles data[2] released by the OPUS project (Tiedemann, 2009).

The experiments targeted both the news-commentary (`nc`) and OpenSubtitles (`st`) domains, with `nc-devtest2007` and `nc-test2007` for tuning and testing in the `nc` domain, respectively, and corresponding 2000 sentence tuning and test sets selected from the `st` data. The news-commentary v7 corpus and a 200k sentence corpus selected from the remaining `st` data were used as in-domain training data for the respective domains, with europarl v7 (`ep`) used as out-of-domain training data in both cases. The language pairs we tested were the WMT language pairs for `nc` (English (en) to and from Spanish (es), German (de), French (fr) and Czech (cs)), with Dutch (nl) substituted for de in the `st` experiments.

To build phrase-based translation systems, we used the standard Moses (Koehn et al., 2007) training pipeline, in particular employing the usual 5 phrase features – forward and backward phrase probabilities, forward and backward lexical scores and a phrase penalty. The 5-gram Kneser-Ney smoothed language models were trained by SRILM (Stolcke, 2002), with KenLM (Heafield, 2011) used at runtime. The language model is always a linear interpolation of models estimated on the in- and out-of-domain corpora, with weights tuned by SRILM's perplexity minimisation[3]. All experiments were run three times with BLEU scores averaged, as recommended by Clark et al. (2011). Performance was evaluated using case-insensitive BLEU (Papineni et al., 2002), as implemented in Moses.

The baseline systems were tuned using the Moses version of PRO, a reimplementation of the original

[2] `www.opensubtitles.org`

[3] Our method could also be applied to language model interpolation but we chose to focus on phrase tables in this paper.

algorithm using the sampling scheme recommended by Hopkins and May. We ran 15 iterations of PRO, choosing the weights that maximised BLEU on the tuning set. For the PRO training of the interpolated models, we used the same sampling scheme, with optimisation of the model weights and interpolation weights implemented in Python using scipy[4]. The implementation is available in Moses, in the `contrib/promix` directory.

The phrase table interpolation and perplexity-based minimisation of interpolation weights used the code accompanying Sennrich (2012), also available in Moses.

3.2 Results

For each of the two test sets (`nc` and `st`), we compare four different translation systems (three baseline systems, and our new interpolation method):

in Phrase and reordering tables were built from just the in-domain data.

joint Phrase and reordering tables were built from the in- and out-of-domain data, concatenated.

perp Separate phrase tables built on in- and out-of-domain data, interpolated using perplexity minimisation. The reordering table is as for **joint**.

pro-mix As **perp**, but interpolation weights optimised using our modified PRO algorithm.

So the two interpolated models (**perp** and **pro-mix**) are the same as **joint** except that their 4 non-constant phrase features are interpolated across the two separate phrase tables. Note that the language models are the same across all four systems.

The results of this comparison over the 8 language pairs are shown in Figure 1, and summarised in Table 1, which shows the mean BLEU change relative to the **in** system. It can be seen that the **pro-mix** method presented here is out-performing the perplexity optimisation on the `nc` data set, and performing similarly on the `st` data set.

	joint	**perp**	**pro-mix**
`nc`	+0.18	+0.44	+0.91
`st`	-0.04	+0.55	+0.48

Table 1: Mean BLEU relative to **in** system for each data set. System names as in Figure 1

[4] `www.scipy.org`

4 Discussion and Conclusions

The results show that the **pro-mix** method is a viable way of tuning systems built with interpolated phrase tables, and performs better than the current perplexity minimisation method on one of two data sets used in experiments. On the other data set (`st`), the out-of-domain data makes much less difference to the system performance in general, most probably because the difference between the in and out-of-domain data sets in much larger (Haddow and Koehn, 2012). Whilst the differences between **pro-mix** and perplexity minimisation are not large on the `nc` test set (about +0.5 BLEU) the results have been demonstrated to apply across many language pairs.

The advantage of the **pro-mix** method over other approaches is that it directly optimises the measure that we are interested in, rather than optimising an intermediate measure and hoping that translation performance improves. In this work we optimise for BLEU, but the same method could easily be used to optimise for any sentence-level translation metric.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 288769 (ACCEPT).

References

Pratyush Banerjee, Sudip K. Naskar, Johann Roturier, Andy Way, and Josef van Genabith. 2011. Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component Level Mixture Modelling. In *Proceedings of MT Summit*.

Nicola Bertoldi and Marcello Federico. 2009. Domain Adaptation for Statistical Machine Translation from Monolingual Resources. In *Proceedings of WMT*.

R. H. Byrd, P. Lu, and J. Nocedal. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012.

Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of NAACL*.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online Large-Margin Training of Syntactic and Structural Translation Features. In *Proceedings of EMNLP*.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL*.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech*, Brisbane, Australie.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA, October. Association for Computational Linguistics.

Barry Haddow and Philipp Koehn. 2012. Analysing the Effect of Out-of-Domain Data on SMT Systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montréal, Canada, June. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as Ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL Demo Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Franz J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Rico Sennrich. 2012. Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of EACL*.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. on Spoken Language Processing, vol. 2*, pages 901–904.

Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing (vol V)*, pages 237–248. John Benjamins, Amsterdam/Philadelphia.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online Large-Margin Training for Statistical Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic, June. Association for Computational Linguistics.

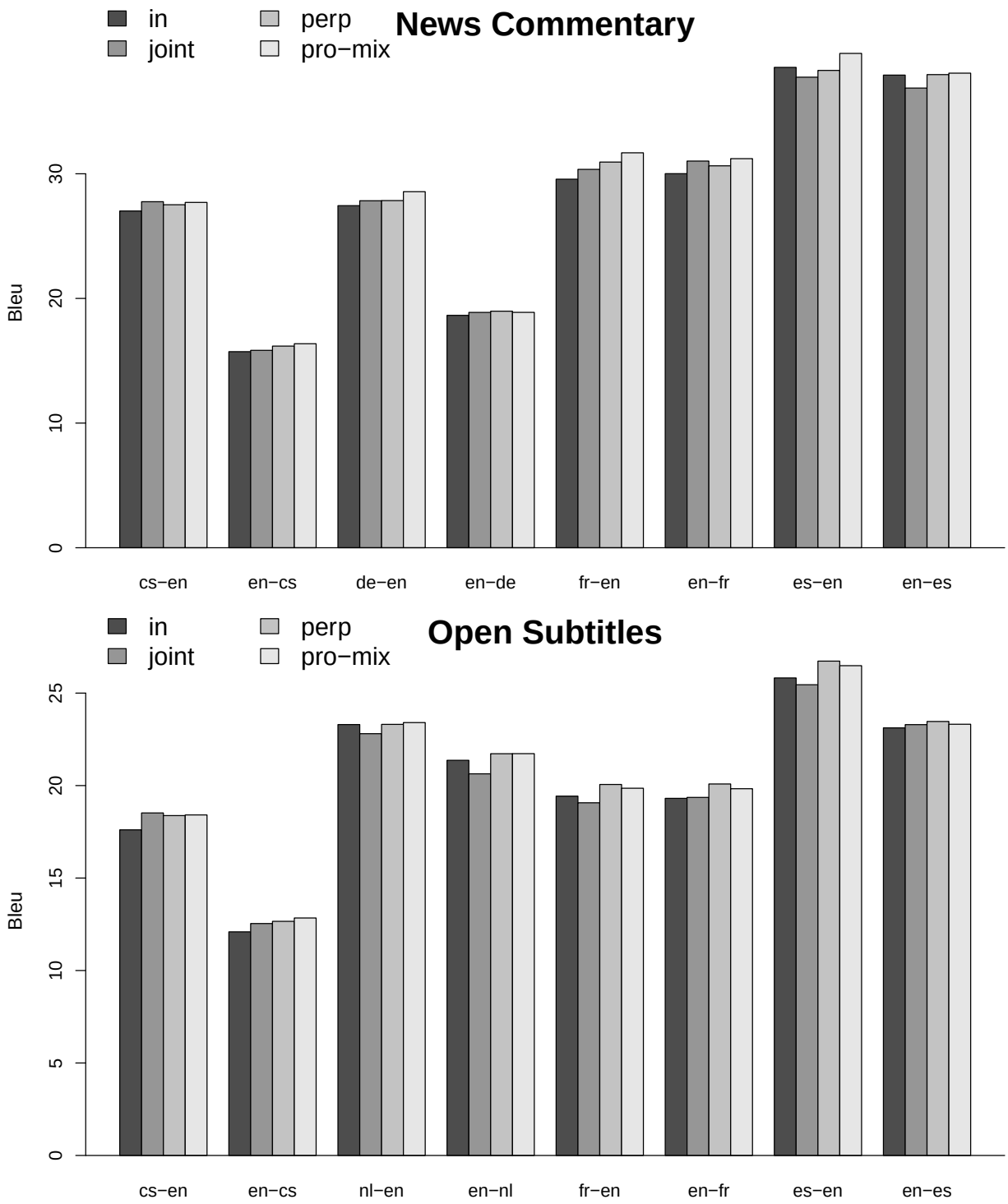


Figure 1: Comparison of the performance (BLEU) on in-domain data, of our **pro-mix** interpolation weight tuning method with three baselines: **in** using just in-domain parallel training data training; **joint** also using europarl data; and **perp** using perplexity minimisation to interpolate in-domain and europarl data.

Analysing the Effect of Out-of-Domain Data on SMT Systems

Barry Haddow and Philipp Koehn
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, Scotland
{bhaddow,pkoehn}@inf.ed.ac.uk

Abstract

In statistical machine translation (SMT), it is known that performance declines when the training data is in a different domain from the test data. Nevertheless, it is frequently necessary to supplement scarce in-domain training data with out-of-domain data. In this paper, we first try to relate the effect of the out-of-domain data on translation performance to measures of corpus similarity, then we separately analyse the effect of adding the out-of-domain data at different parts of the training pipeline (alignment, phrase extraction, and phrase scoring). Through experiments in 2 domains and 8 language pairs it is shown that the out-of-domain data improves coverage and translation of rare words, but may degrade the translation quality for more common words.

1 Introduction

In statistical machine translation (SMT), domain adaptation can be thought of as the problem of training a system on data mainly drawn from one domain (e.g. parliamentary proceedings) and trying to maximise its performance on a different domain (e.g. news). There is likely to be some parallel data similar to the test data, but as such data is expensive to create, it tends to be scarce. The concept of "domain" is rarely given a precise definition, but it is normally understood that data from the same domain is in some sense similar (for example in the words and grammatical constructions used) and data from different domains shows less similarities. Data from the same domain as the test set is usually referred to as *in-domain* and data from a different domain is referred to as *out-of-domain*.

The aim of this paper is to shed some light on what domain actually is, and why it matters. The fact that a mismatch between training and test data domains reduces translation performance has been observed in previous studies, and will be confirmed here for multiple data sets and languages, but the question of why domain matters for performance has not been fully addressed in the literature. Experiments in this paper will be conducted on phrase-based machine translation (PBMT) systems, but similar conclusions are likely to apply to other types of SMT systems. Furthermore, we will mainly be concerned with the effect of domain on the translation model, since it depends on parallel data which is more likely to be in short supply than monolingual data, and domain adaptation for language modelling has been more thoroughly studied.

The effect of a shift of domain in the parallel data is complicated by the fact that training a translation model is a multi-stage process. First the parallel data is word-aligned, normally using the IBM models (Brown et al., 1994), then phrases are extracted using some heuristics (Och et al., 1999) and scored using a maximum likelihood estimate. Since the effect of domain may be felt at the alignment stage, the extraction stage, or the scoring stage, we have designed experiments to try to tease these apart. Experiments comparing the effect of domain at the alignment stage with the extraction and scoring stages have already been presented by (Duh et al., 2010), so we focus more on the differences between extraction and scoring. In other words, we examine whether adding more data (in or out-of domain) helps improve coverage of the phrase table, or helps improve the scoring of phrases.

A further question that we wish to address is

whether adding out-of-domain parallel data affects words with different frequencies to different degrees. For example, a large out-of-domain data set may improve the translation of rare words by providing better coverage, but degrade translation of more common words by providing erroneous out-of-domain translations. In fact, the evidence presented in Section 3.5 will show a much clearer effect on low frequency words than on medium or high frequency words, but the total token count of these low frequency words is still small, so they don't necessarily have much effect on overall measures of translation quality.

In summary, the main contributions of this paper are:

- It presents experiments on 8 language pairs and 2 domains showing the effect on BLEU of adding out-of-domain data.
- It provides evidence that the difference between in and out-of domain translation performance is correlated with differences in word distribution and out-of-vocabulary rates.
- It develops a method for separating the effects of phrase extraction and scoring, showing that good coverage is nearly always more important than good scoring, and that out-of-domain data can adversely affect phrase scores.
- It shows that adding out-of-domain data clearly improves translation of rare words, but may have a small negative effect on more common words.

2 Related Work

The most closely related work to the current one is that of (Duh et al., 2010). In this paper they consider the domain adaptation problem for PBMT, and investigate whether the out-of-domain data helps more at the word alignment stage, or at the phrase extraction and scoring stages. Extensive experiments on 4 different data sets, and 10 different language pairs show mixed results, with the overall conclusion being that it is difficult to predict how best to include out-of-domain data in the PBMT training pipeline. Unlike in the current work, Duh et al. do not separate phrase extraction and scoring in order to analyse the effect of domain on them separately. They make the point that adding extra out-of-domain data may degrade translation by introducing unwanted lexical ambiguity, showing anecdotal evidence for this. Similar arguments were presented in (Sennrich, 2012).

A recent paper which does attempt to tease apart phrase extraction and scoring is (Bisazza et al., 2011). In this work, the authors try to improve a system trained on in-domain data by including extra entries (termed "fill-up") from out-of-domain data – this is similar to the `nc+epE` and `st+epE` systems in Section 3.4. It is shown by Bisazza et al. that this fill-up technique has a similar effect to using MERT to weight the in and out-of domain phrase tables. In the experiments in Section 3.4 we confirm that fill-up techniques mostly provide better results than using a concatenation of in and out-of domain data.

There has been quite a lot of work on finding ways of weighting in and out-of domain data for SMT (as opposed to simply concatenating the data sets), both for language and translation modelling. Interpolating language models using perplexity is fairly well-established (e.g. Koehn and Schroeder (2007)), but for phrase-tables it is unclear whether perplexity minimisation (Foster et al., 2010; Sennrich, 2012) or linear or log-linear interpolation (Foster and Kuhn, 2007; Civera and Juan, 2007; Koehn and Schroeder, 2007) is the best approach. Also, other authors (Foster et al., 2010; Niehues and Waibel, 2010; Shah et al., 2010) have tried to weight the input sentences or extracted phrases before the phrase tables are built. In this type of approach, the main problem is how to train the weights of the sentences or phrases, and each of the papers has followed a different approach.

Instead of weighting the out-of-domain data, some authors have investigated data selection methods for domain adaptation (Yasuda et al., 2008; Mansour et al., 2011; Schwenk et al., 2011; Axelrod et al., 2011). This is effectively the same as using a 1-0 weighting for input sentences, but has the advantage that it is usually easier to tune a threshold than it is to train weights for all input sentences or phrases. The other advantage of doing data selection is that it can potentially remove noisy (e.g. incorrectly aligned) data. However it will be seen later in this paper that out-of-domain data can usually contribute something useful to the translation system, so the 1-0 weighting of data-selection may be somewhat heavy-handed.

3 Experiments

3.1 Corpora and Baselines

The experiments in this paper used data from the WMT09 and WMT11 shared tasks (Callison-Burch et al., 2009; Callison-Burch et al., 2011), as well as OpenSubtitles data[1] released by the OPUS project (Tiedemann, 2009).

From the WMT data, both news-commentary-v6 (`nc`) and europarl-v6 (`ep`) were used for training translation models and language models, with `nc-devtest2007` used for tuning and `nc-test2007` for testing. The experiments were run on all language pairs used in the WMT shared tasks, i.e. English (en) into and out of Spanish (es), German (de), French (fr) and Czech (cs).

From the OpenSubtitles (`st`) data, we chose 8 language pairs – English to and from Spanish, French, Czech and Dutch (nl) – selected because they have at least 200k sentences of parallel data available. 2000 sentence tuning and test sets (`st-dev` and `st-devtest`) were selected from the parallel data by extracting every nth sentence, and a 200k sentence training corpus was selected from the remaining data.

Using test sets from both news-commentary and OpenSubtitles gives two domain adaptation tasks, where in both cases the out-of-domain data is europarl, a significantly larger training set than the in-domain data. The three data sets in use in this paper are summarised in Table 1.

The translation systems consisted of phrase tables and lexicalised reordering tables estimated using the standard Moses (Koehn et al., 2007) training pipeline, and 5-gram Kneser-Ney smoothed language models estimated using the SRILM toolkit (Stolcke, 2002), with KenLM (Heafield, 2011) used at runtime. Separate language models were built on the target side of the in-domain and out-of-domain training data, then linearly interpolated using SRILM to minimise perplexity on the tuning set (e.g. Koehn and Schroeder (2007)). Tuning of models used minimum error rate training (Och, 2003), repeated 3 times and averaged (Clark et al., 2011). Performance is evaluated using case-insensitive BLEU (Papineni et al., 2002), as implemented using the Moses `multi-bleu.pl` script.

[1] `www.opensubtitles.org`

Name	Language pairs	train	tune	test
Europarl (`ep`)	en↔fr	1.8M	n/a	n/a
	en↔es	1.8M	n/a	n/a
	en↔de	1.7M	n/a	n/a
	en↔cs	460k	n/a	n/a
	en↔nl	1.8M	n/a	n/a
News Commentary (`nc`)	en↔fr	114k	1000	2000
	en↔es	130k	1000	2000
	en↔de	135k	1000	2000
	en↔cs	122k	1000	2000
Subtitles (`st`)	en↔fr	200k	2000	2000
	en↔es	200k	2000	2000
	en↔nl	200k	2000	2000
	en↔cs	200k	2000	2000

Table 1: Summary of the data sets used, with approximate sentence counts

3.2 Comparing In-domain and Out-of-domain Data

The aim of this section is to provide both a qualitative and quantitative comparison of the three data sets used in this paper.

Firstly, consider the extracts from the English sections of the three training sets shown in Figure 1. The first extract, from the Europarl corpus, shows a formal style with long sentences. However this is still spoken text so contains a preponderance of first and second person forms. In terms of subject matter, the corpus covers a broad range of topics, but all from the angle of European legislation, institutions and policies. Where languages (e.g. English, French and Spanish) have new world and old world variants, Europarl sticks to the old world variants.

The extract from the News Commentary corpus again shows a formal tone, but because this is news analysis, it tends to favour the third person. It is written text, and covers a wider range of subjects than Europarl, and also encompasses both new and old world versions of the European languages.

The Subtitles text shown in the last example appears qualitatively more different from the other two. It is spoken text, like Europarl, but consists of short, informal sentences with many colloquialisms, as well as possible optical character recognition er-

Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful.
You have requested a debate on this subject in the course of the next few days, during this part-session.
In the meantime, I should like to observe a minute' s silence, as a number of Members have requested, on behalf of all the victims concerned, particularly those of the terrible storms, in the various countries of the European Union.

(a) Europarl

Desperate to hold onto power, Pervez Musharraf has discarded Pakistan's constitutional framework and declared a state of emergency.
His goal?
To stifle the independent judiciary and free media.
Artfully, though shamelessly, he has tried to sell this action as an effort to bring about stability and help fight the war on terror more effectively.

(b) News commentary

I'il call in 30 minutes to check
Is your mother here, too?
Why are you outside?
It's no fun listening to women's talk
Well, why don't we go in together

(c) OpenSubtitles

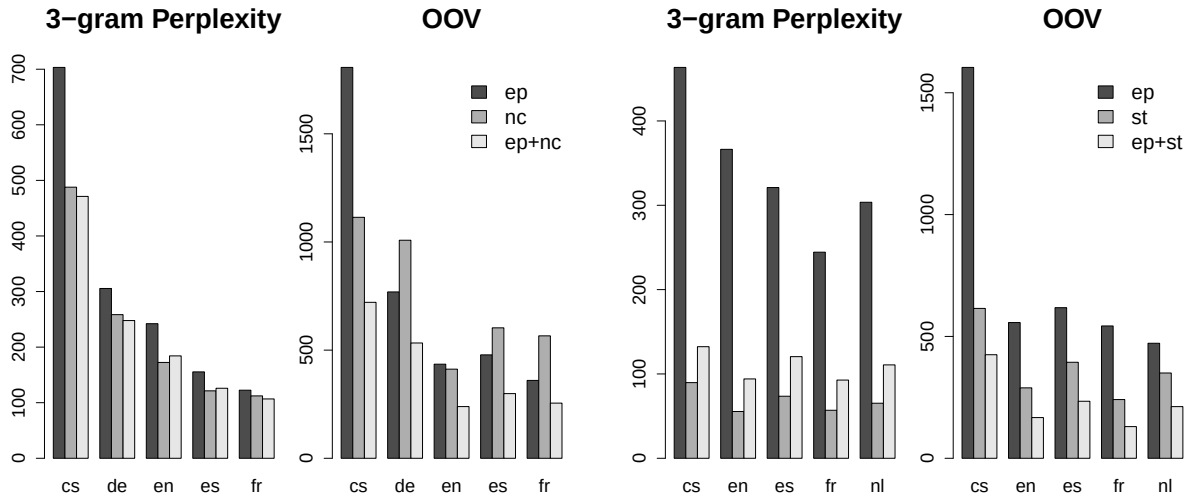
Figure 1: Extracts from the English portion of the training corpora

rors. It is likely to contain a mixture of regional variations of the languages, reflecting the diversity of the film sources.

In order to obtain a quantitative measure of domain differences, we used both language model (LM) perplexity, and out-of-vocabulary (OOV) rate, in the two test domains. For the `nc` domain, perplexity was compared by training trigram LMs (with SRILM and Kneser-Ney smoothing) on each of the `ep`, `nc` and `ep+nc` training sets, taking the intersection of the `ep` and `nc` vocabularies as the LM vocabulary. The perplexities of the `nc` test set wer calculated using each of the LMs. A corresponding set of LMs was trained to compare perplexities on the `st` test set, and all perplexity comparisons were performed on all five languages. The SRILM toolkit was also used to calculate OOV rates on the test set, by training language models with an open vocabulary, and using no unknown word probability estimation.

The perplexities and OOV rates on each test corpora are shown in Figure 2. The pattern of perplexities is quite distinct across the two test domains, with the perplexity from out-of-domain data relatively much higher for the `st` test set. The in-domain data LM also shows the lowest perplexity consistently on this test set, whilst for `nc`, the in-domain LM has a similar perplexity to the `ep+nc` LM. In fact for 3/5 languages (fr,cs and de) the `ep+nc` LM has the lowest perplexity.

With regard to the OOV rates, it is notable that for `nc` the rate is actually higher for the in-domain LM than the out-of-domain LM in three of the languages: French, German and Spanish. The most likely reason for this is that these languages have a relatively rich morphology, so the larger out-of-domain corpus (Table 1) gives greater coverage of the different grammatical suffixes. Czech shows a different pattern because in this case the out-of-domain corpus is not much bigger than the in-domain corpus, and English is morphologically much simpler so the increase in corpus size does not help the OOV rate so much.



(a) Test on news commentary.

(b) Test on subtitles.

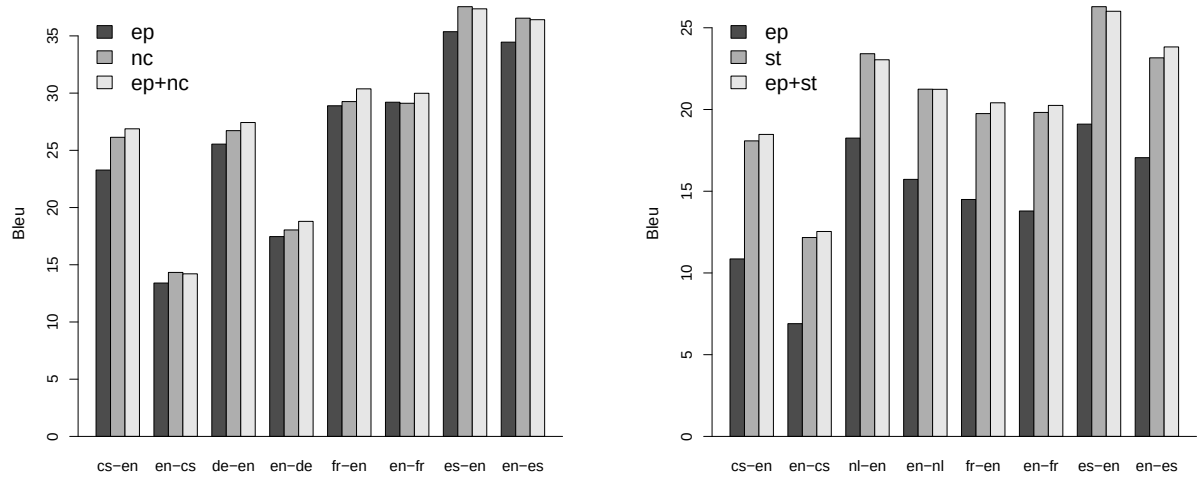
Figure 2: Comparison of perplexities and OOV rates on in-domain test data

3.3 Comparing Translation Performance of In and Out-of-domain Systems

Translation performance was measured on each of the test sets (`nc` and `st`) using systems built from just the in-domain parallel data, from just the out-of-domain parallel data, and on a concatenation of the in and out-of domain data. In other words, systems built from the `ep`, `nc` and `ep+nc` parallel texts were evaluated on the `nc` test data, and systems built from `ep`, `st` and `ep+st` were evaluated on the `st` test data. In all cases, the parallel training set was used to build both the phrase table and the lexicalised reordering models, the language model was the interpolated one described in Section 3.1, and the system was tuned on data from the same domain as the test set.

From Figure 3 it is clear that the difference between the in and out-of domain training sets is much bigger for `st` than for `nc`. The BLEU scores on `nc` for the `nc` trained systems are on average 1.3 BLEU points higher than those for the `ep` trained systems, whilst the scores on `st` gain an average of 6.0 BLEU points when the training data is switched from `ep` to `st`. The patterns are quite consistent across languages for the `st` tested systems, with the gains varying just from 5.2 to 7.2. However for the `nc` tested systems there are some language pairs which show a gain of more than 2 BLEU points when moving from out-of to in-domain training data (cs-en, en-es and es-en), whereas en-fr shows no change. The main link between the perplexity and OOV results in Figure 2 and the BLEU score variations in Figure 3 is that the larger in/out differences between the two domains is reflected in larger BLEU differences. However it is also notable that the two languages which display a rise in perplexity between `nc` and `ep+nc` are es and en, and for both es-en and en-es the `ep+nc` translation system performs worse than the `nc` trained system.

The BLEU gain from concatenating the in and out-of domain data, over just using the in-domain data can be quite small. For the `nc` domain this averages at 0.5 BLEU (with 3/8 language pairs showing a decrease), whilst for the `st` domain the average gain is only 0.2 BLEU (with again 3/8 language pairs showing a decrease). So even though adding the out-of-domain data increases the training set size by a factor of 10 in most cases, its effect on BLEU score is small.



(a) Test on news commentary

(b) Test on subtitles

Figure 3: Comparison of translation performance using models from in-domain, out-of-domain and joint data.

3.4 Why Does Adding Parallel Data Help?

In the previous section it was found that, across all language pairs and both data sets, adding in-domain data to an out-of-domain training set nearly always has a positive impact on performance, whilst adding out-of-domain data to an in-domain training set can sometimes have a small positive effect. In this section several experiments are performed with "intermediate" phrase tables (built from a single parallel corpus, augmented with some elements of the other parallel corpus) in order to determine how different aspects of the extra data affect performance. In particular, the experiments are designed to show the effect of the extra data on the alignments, the phrase scoring and the phrase coverage, whether adding in-domain data to an existing out-of-domain trained system, or vice-versa.

For each of the language pairs used in this paper, and each of the two domains, two series of experiments were run comparing systems built from a single parallel training set, intermediate systems, and systems built from a concatenation of the in and out-of-domain parallel data sets. Only the parallel data was varied, the language models were as described in Section 3.1, and the lexicalised reordering models were built from both training sets in all cases, except for the systems built from a single parallel data set[2]. This gives a total of four series of experiments, where the ordered pair of data sets (`x`,`y`) was set to one of (`ep`,`nc`), (`nc`,`ep`), (`ep`,`st`), (`st`,`ep`). In each of these series, the following translation systems were trained:

- `x` The translation table and lexicalised reordering model were estimated from the `x` corpus alone.
- `x+y` The translation system built from the `x` and `y` parallel corpora concatenated.
- `x+yA` As `x` but using the additional `y` corpus to create the alignments. This means that GIZA++ was run across the entire `x+y` corpus, but only the `x` section of it was used to extract and score phrases.
- `x+yW` As `x+yA` but using the phrase scores from the `x+y` phrase table. This is effectively the `x+y` system, with any entries in the phrase table that are just found in the `y` corpus removed.

[2]Further experiments were run using the parallel data from a single data set to build the translation model, and both data sets to build the lexicalised reordering model, but the difference in score compared to the `x` system was small (< 0.1 BLEU)

x+yE As x+yA but adding the extra entries from the x+y phrase table. This is effectively the x+y system, but with the scores on all phrases that are found in x phrase table set to their values from that table.

All systems were tuned and tested on the appropriate in-domain data set (either nc or st). Note that in the intermediate systems, the phrase table scores may no longer correspond to valid probability distributions, but this is not important as the probabilistic interpretation is never used in decoding anyway.

The graphs in Figure 4 show the performance comparison between the single corpus systems, the intermediate systems, and the concatenated corpus systems, averaged across all 8 language pairs. Table 2 shows the full results broken down by language pair, for completeness, but the patterns are reasonably consistent across language pair.

Firstly, compare the x+yW and x+yE systems, i.e. the systems where we add just the weights from the second parallel data set versus those where we add just the entries. When x is the out-of-domain (ep) data, then it is clearly more profitable to update the phrase-table entries than the weights from the in-domain data. In fact for the systems tested on st, the difference is quite striking with a $+5.7$ BLEU gain for the ep+stE system over the baseline ep system, but only a $+1.5$ gain for the ep+stW system. For the systems tested on the nc, adding the entries from nc gives a larger gain in BLEU than adding the weights ($+1.3$ versus $+0.8$), but both improve the BLEU scores over the ep+ncA system. The conclusion is that the extra entries from the in-domain data (the "fill-up" of Bisazza et al. (2011)) are more important than the improvements in phrase scoring that in-domain data may provide.

Looking at the other two sets of x+yW and x+yE systems, i.e. those where x is the in-domain data, tells another story. In this case, the results on both the nc and st test sets (Figure 4(b)) suggest that it is generally more useful to use the out-of-domain data as only a source of extra phrase-table entries. This is because the x+epE systems are the highest scoring in both cases, scoring higher than systems built from all the data concatenated by margins of 0.5 (for nc) and 0.4 (for st). This pattern is consistent across all the language pairs for nc, and across 5 of the 8 language pairs for st. Using the out-of-domain data set to update only the weights (the x+epW systems) generally degrades performance when compared to the systems that only use the ep data at alignment time (the x+epA systems).

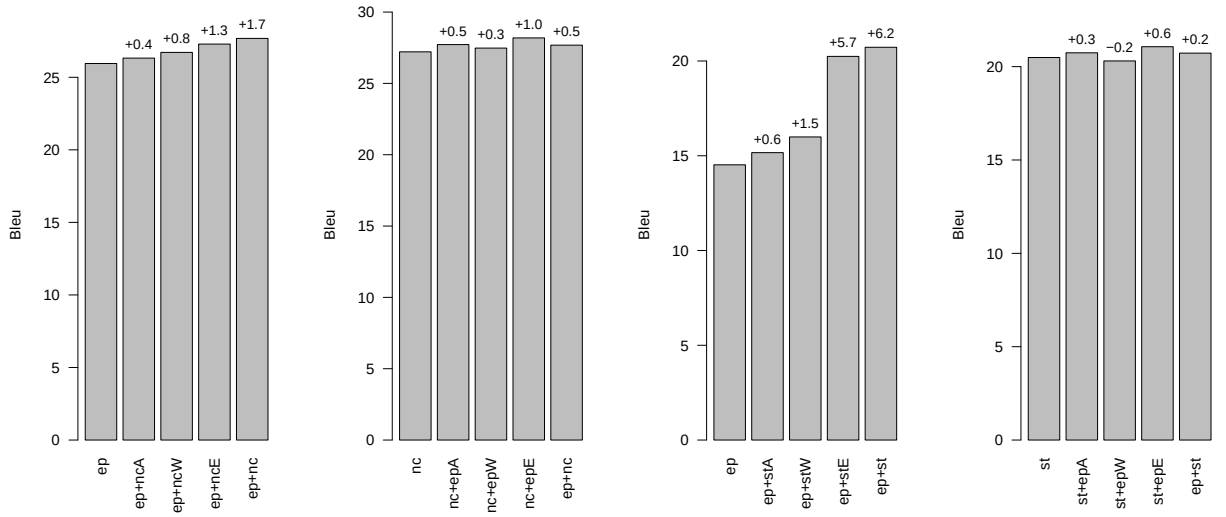
The size of the effect of adding extra data to the alignment stage only is mixed (as observed by (Duh et al., 2010)), but in general all the x+yA systems show an improvement over the x systems. In fact, for the st domain, adding ep at the alignment stage is the only consistent way to improve BLEU. Adding the weights, entries, or the complete out-of-domain data set does not always help.

3.5 Word Precision Versus Frequency

The final set of experiments addresses the question of whether the change of translation quality when adding out-of-domain has a different effect depending on word frequency. To do this, the systems trained on in-domain only are compared with the systems trained on all data concatenated, using a technique for measuring the precision of the translation for each word type.

To calculate the precision of a word type, it is necessary to examine each translated sentence to see which source words were translated correctly. This is done by recording the word alignment in the phrase mappings and tracking it through the translation process. If a word is produced multiple times in the translation, but occurs a fewer number of times in the reference, then it is assigned partial credit. Many-to-many word alignments are treated similarly. Precision for each word type is then calculated in the usual way, as the number of times that word appears correctly in the output, divided by the total number of appearances. The word types are then binned according to the log2 of their frequency in the in-domain corpus and the average precision for each bin calculated, then these are in turn averaged across language pairs.

The graphs in Figure 5 compare the in-domain source frequency versus precision relationship for systems built using just the in-domain data, and systems built using both in and out-of domain data. There is a consistent increase in precision for lower frequency words (occurring less than 30 times in training), but the total number of occurrences of these words is low, so they contribute less to over-

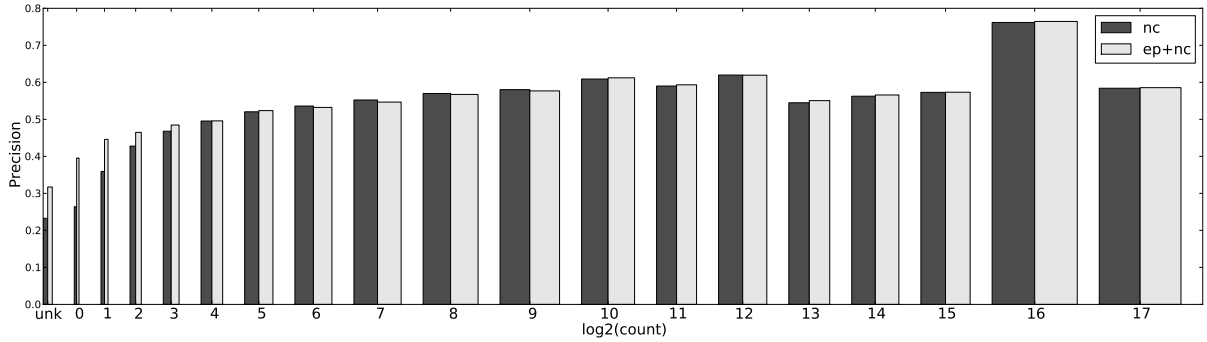


(a) Start with `ep`, test on `nc`. (b) Start with `nc`, test on `nc`. (c) Start with `ep`, test on `st`. (d) Start with `st`, test on `st`

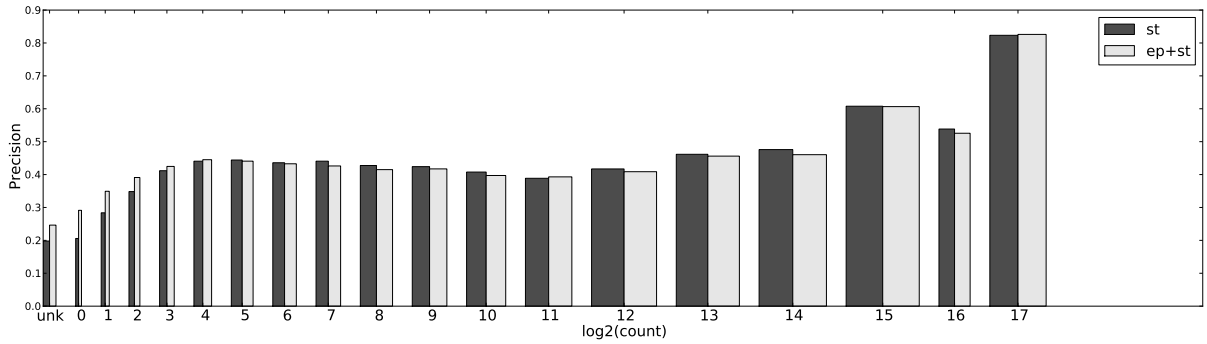
Figure 4: Showing the performance change when starting with either in or out-of domain data, and adding elements of the other data set. The "A" indicates that the second data set is only used for alignments, the "W" indicates that it contributes alignments and phrase scores, and the "E" indicates that it contributes alignments and phrase entries. The figures above each bar shows the performance change relative to the single corpus system.

System	cs-en	en-cs	de-en	en-de	fr-en	en-fr	es-en	en-es
ep	23.3	13.4	25.5	17.5	28.9	29.2	35.4	34.5
ep+ncA	23.5 (+0.2)	13.8 (+0.4)	25.9 (+0.4)	17.9 (+0.4)	29.3 (+0.4)	29.6 (+0.4)	35.7 (+0.3)	34.9 (+0.5)
ep+ncW	24.0 (+0.7)	14.2 (+0.8)	26.3 (+0.8)	18.2 (+0.7)	29.4 (+0.5)	29.8 (+0.6)	36.3 (+0.9)	35.6 (+1.1)
ep+ncE	26.2 (+2.9)	14.0 (+0.6)	27.0 (+1.5)	18.5 (+1.0)	29.7 (+0.9)	30.0 (+0.8)	37.0 (+1.7)	35.7 (+1.3)
nc	26.1 (+2.9)	14.3 (+0.9)	26.7 (+1.2)	18.0 (+0.6)	29.3 (+0.4)	29.1 (-0.1)	37.6 (+2.2)	36.5 (+2.1)
nc+epA	26.8 (+3.5)	14.6 (+1.2)	27.5 (+2.0)	18.5 (+1.0)	30.4 (+1.5)	29.9 (+0.7)	37.7 (+2.3)	36.4 (+2.0)
nc+epW	26.6 (+3.3)	14.4 (+1.0)	27.4 (+1.9)	18.4 (+1.0)	29.5 (+0.6)	29.8 (+0.6)	37.2 (+1.8)	36.5 (+2.0)
nc+epE	**27.4 (+4.1)**	**14.7 (+1.3)**	**28.1 (+2.6)**	**19.0 (+1.5)**	**30.9 (+2.0)**	**30.2 (+1.0)**	**38.4 (+3.0)**	**36.9 (+2.4)**
ep+nc	26.9 (+3.6)	14.2 (+0.8)	27.4 (+1.9)	18.8 (+1.3)	30.4 (+1.5)	30.0 (+0.8)	37.4 (+2.0)	36.4 (+2.0)
System	cs-en	en-cs	nl-en	en-nl	fr-en	en-fr	es-en	en-es
ep	10.9	6.9	18.2	15.7	14.5	13.8	19.1	17.1
ep+stA	11.9 (+1.0)	7.5 (+0.6)	19.0 (+0.8)	16.3 (+0.5)	15.0 (+0.5)	14.1 (+0.3)	19.8 (+0.7)	17.8 (+0.7)
ep+stW	12.2 (+1.3)	8.1 (+1.2)	20.0 (+1.7)	17.4 (+1.7)	15.8 (+1.3)	14.9 (+1.1)	20.8 (+1.7)	18.8 (+1.8)
ep+stE	18.0 (+7.1)	12.4 (+5.5)	22.5 (+4.2)	20.6 (+4.9)	19.6 (+5.1)	19.9 (+6.1)	25.6 (+6.5)	23.3 (+6.3)
st	18.0 (+7.2)	12.2 (+5.3)	23.4 (+5.1)	21.3 (+5.6)	19.7 (+5.2)	19.8 (+6.0)	26.3 (+7.2)	23.2 (+6.1)
st+epA	18.4 (+7.6)	12.4 (+5.5)	23.6 (+5.4)	21.3 (+5.6)	20.2 (+5.7)	20.1 (+6.3)	**26.4 (+7.3)**	23.5 (+6.5)
st+epW	18.2 (+7.3)	12.2 (+5.3)	22.4 (+4.2)	21.0 (+5.3)	19.9 (+5.4)	19.8 (+6.0)	25.8 (+6.7)	23.2 (+6.1)
st+epE	**19.1 (+8.3)**	12.5 (+5.6)	**24.0 (+5.8)**	**21.7 (+6.0)**	**20.6 (+6.1)**	**20.9 (+7.1)**	26.0 (+6.9)	23.7 (+6.6)
ep+st	18.5 (+7.6)	**12.5 (+5.6)**	23.0 (+4.8)	21.2 (+5.5)	20.4 (+5.9)	20.2 (+6.5)	26.0 (+6.9)	**23.8 (+6.8)**

Table 2: Complete scores for the experiments described in Section 3.4 and summarised in Figure 4. Naming of the systems is explained in the text, and in the caption for Figure 4



(a) News commentary



(b) Subtitles

Figure 5: Performance comparison of in-domain systems versus systems built from in and out-of domain data concatenated. Precision is plotted against log2 of in-domain training frequency, and averaged across all 8 language pairs. The width of the bars indicates the average total number of occurrences in the test set.

all measures of translation quality. For the words with moderate training set frequencies, the precision is actually slightly higher for the systems built with just in-domain data, an effect that is more marked for the `st` domain.

4 Conclusions

In this paper we have attempted to give an in-depth analysis of the domain adaptation problem for two different domain adaptation problems in phrase-based MT. The differences between the two problems are clearly illustrated by the results in Figures 2 and 3, where we see that the difference between the in-domain and out-of-domain data are larger for the OpenSubtitles domain than for the News-Commentary domain. This can be detected by the differences in word distribution and out-of-vocabulary rates observed in Figure 2, and is reflected by the differing translation results in Figure 3.

However, the experiments of Sections 3.4 and 3.5 show some common themes emerging in the two domains. In both cases, the out-of-domain data helps most when it is just allowed to add entries (i.e. "fill in") the phrase-table, and using the scores provided by out-of-domain data has a tendency to be harmful to translation quality. The precision results of Section 3.5 show out-of-domain data (when it is simply added to the training set) mainly helping with the low frequency words, and having a neutral or harmful effect for higher frequency words. This explains why approaches which try to weight the out-of-domain data in some way (e.g. corpus weighting or instance weighting) can be more successful than

simply concatenating data sets. It also suggests that the way forward is to look for methods that use the out-of-domain data mainly for rarer words, and not to change translations which have a lot of evidence in the in-domain data.

5 Acknowledgments

This work was supported by the EuroMatrixPlus[3] and Accept[4] projects funded by the European Commission (7th Framework Programme).

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-Domain Data Selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *Proceedings of IWSLT*.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1994. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece, March. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

Jorge Civera and Alfons Juan. 2007. Domain Adaptation in Statistical Machine Translation with Mixture Modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL*.

[3] www.euromatrixplus.net

[4] www.accept.unige.ch

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation in statistical machine translation. In *Proceedings of IWSLT*.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA, October. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL Demo Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Saab Mansour, Joern Wuebker, and Hermann Ney. 2011. Combining Translation and Language Model Scoring for Domain-Specific Data Filtering. In *Proceedings of IWSLT*.

Jan Niehues and Alex Waibel. 2010. Domain Adaptation in Statistical Machine Translation using Factored Translation Models. In *Proceedings of EAMT*.

Franz J. Och, Christoph Tillman, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Franz J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th*

Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Holger Schwenk, Patrik Lambert, Loïc Barrault, Christophe Servan, Sadaf Abdul-Rauf, Haithem Afli, and Kashif Shah. 2011. LIUM's SMT Machine Translation Systems for WMT 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 464–469, Edinburgh, Scotland, July. Association for Computational Linguistics.

Rico Sennrich. 2012. Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of EACL*.

Kashif Shah, Loïc Barrault, and Holger Schwenk. 2010. Translation Model Adaptation by Resampling. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 392–399, Uppsala, Sweden, July. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. on Spoken Language Processing, vol. 2*, pages 901–904.

Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing (vol V)*, pages 237–248. John Benjamins, Amsterdam/Philadelphia.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of Selecting Training Data to Build a Compact and Efficient Translation Model. In *Proceedings of IJCNLP*.

Sparse Lexicalised Features and Topic Adaptation for SMT

Eva Hasler, Barry Haddow, Philipp Koehn

University of Edinburgh
Edinburgh, United Kingdom
e.hasler@ed.ac.uk, {pkoehn,bhaddow}@inf.ed.ac.uk

Abstract

We present a new approach to domain adaptation for SMT that enriches standard phrase-based models with lexicalised word and phrase pair features to help the model select appropriate translations for the target domain (TED talks). In addition, we show how source-side sentence-level topics can be incorporated to make the features differentiate between more fine-grained topics within the target domain (topic adaptation). We compare tuning our sparse features on a development set versus on the entire in-domain corpus and introduce a new method of porting them to larger mixed-domain models. Experimental results show that our features improve performance over a MIRA baseline and that in some cases we can get additional improvements with topic features. We evaluate our methods on two language pairs, English-French and German-English, showing promising results.

1. Introduction

In the field of statistical machine translation, domain adaptation is the task of tuning machine translation systems to produce optimal translations for a particular target domain by making the best possible use of the training data, given that we have, usually, a small amount of in-domain data and a larger amount of out-of-domain data. Most approaches to domain adaptation concentrate on either the language model or the translation model and ways to get more appropriate estimates for the respective probability distributions. Other approaches focus on acquiring more in-domain data as opposed to trying to make better use of existing training data.

In this paper, we focus on enhancing standard phrase-based machine translation systems with sparse features in order to bias our systems for the vocabulary and style of the target domain, the TED talks domain. We explore and compare several discriminative training approaches to include sparse features into small in-domain and larger mixed-domain systems. The idea is that sparse features can be added on top of baseline systems that are trained in the usual fashion, overlapping with existing features in the phrase table. This gives us flexibility to explore new feature sets which is particularly useful for training large systems from mixed-domain data. We show experimental results on data provided for the IWSLT 2012 shared task.

2. Training sparse features for domain adaptation

Adding sparse, lexicalised features to existing translation systems trained on in-domain or mixed-domain data is one way to bias translation systems towards translating a particular domain, in our case the TED talks domain. Our features are trained with the MIRA algorithm which is explained briefly in the following subsection. We compare the standard approach, e.g. tuning on a rather small development set, to the less common jackknife approach, details of which are given in subsection 2.4.

2.1. Training features with MIRA

Recently, the Margin Infused Relaxed Algorithm (MIRA) [6] has gained popularity as an alternative training method to Minimum Error Rate Training (MERT) [16], because it can deal with an arbitrary number of features. MIRA is an online large margin algorithm that enforces a margin between different translations of the same sentence. This margin can be tied to a loss function like BLEU [17] or another quality measure. Given that we can provide the learning algorithm with good oracle translations, the model learns to score hypothesis translations with higher BLEU scores better than translations with lower BLEU scores. MIRA updates the feature weights of a translation model by iterating though the training data, decoding one sentence at a time and performing weight updates for pairs of good and bad translation examples. Details about MIRA can be found in [12] or [3], for example.

We use a slightly modified version of the implementation described in [12] that selects hope and fear translations from a 30best list instead of running the decoder with hope and fear objectives. This has the effect that there is no need for dynamically computed sentence-level BLEU scores anymore because real sentence-level BLEU scores can be computed on the 30best list. [5] mentions that certain features, e.g. the language model, are very sensitive to larger weight changes and so we introduce a separate learning rate for core features (translation model, language model, word penalty and so on) in order to reduce fluctuations and keep MIRA training more stable. This learning rate is independent of the C parameter in the objective function solved by MIRA and is set to 0.1 for core features (1.0 for sparse features).

2.2. Feature sets

We experiment with two classes of indicator features, sparse phrase pair features and sparse word pair (or word translation) features. Word pair features capture translations of single source words to single target words, whereas phrase pair features capture translations of several words on the source side into several words on the target side. The class of phrase pair features depends on the decoder segmentation and can also include phrase pairs of length 1 on each side if such a phrase pair was extracted from the training data. Word pair features on the other hand depend on word alignment information and only contain word pairs that were connected by an alignment point in the training data.

Both of these feature classes were also extended with topic information acquired from topic models trained on the source side of the training corpus. The topic information is integrated as a source side trigger for a particular word or phrase pair, given a topic. Details about how these topic models were trained are given in section 2.3. Table 1 shows a pair of source sentence and hypothesis translation taken from a MIRA training run and examples of the features extracted from that sentence pair. The feature values indicate the number of times a feature occurred in a given sentence pair. The features in the first column capture general word or phrase translations while the features in the second column capture translations given a particular topic (here: topic 10). The features without topic information simply indicate whether a particular word or phrase translation should be favoured or avoided by the decoder, depending on whether they receive positive or negative weights during training. The features with topic information are triggered by the topic of the source sentence, that is, for a particular source sentence to be translated, only the features that were seen with the topic of that sentence will fire.

The TED domain is an interesting domain to try out these classes of features, because we can distinguish two different adaptation tasks: (1) adapting to the general vocabulary of TED talks as opposed to the vocabulary of out-of-domain texts (details in the experiments section), and (2) adapting to the vocabulary of subsets of TED talks that can be grouped into more fine-grained topics which we try to capture with topic models.

2.3. Training topic models

The topic models used for building enhanced word pair and phrase pair features are Hidden Topic Markov Models (HTMMs) [11] and were trained with a freely available toolkit. While topic modelling approaches like Latent Dirichlet Allocation assume that each word in a text was generated by a hidden topic and the topics of all words are assumed to be independent, HTMMs model the topics of words in a document as a Markov chain where all words in a sentence are assigned the same topic. This makes intuitively more sense than assigning several different topics within the same sentence and [11] show that HTMMs also yield lower model perplexity than LDA. The former characteristic makes HTMMs particularly suitable for our purpose. We are guaranteed that each word in a source phrase is assigned the same topic and therefore we do not have to figure out how to assign phrase topics given word topics.

Table 1: *Examples of en-fr word pair (wp) and phrase pair (pp) features, with and without topic information. Brackets indicate the phrase segmentation during decoding.*

input (topic 10): "[a language] [is a] [flash of] [the human spirit] [.]"	
hypothesis: "[une langue] [est une] [flash de] [l' esprit humain] [.] "	
reference: "une langue est une étincelle de l' esprit humain ."	
wp_a∼une=2	wp_10_a∼une=2
wp_language∼langue=1	wp_10_language∼langue=1
wp_is∼est=1	wp_10_is∼est=1
wp_flash∼ flash=1	wp_10_flash∼ flash=1
wp_of∼de=1	wp_10_of∼de=1
...	...
pp_a,language∼une,langue=1	pp_10_a,language∼une,langue=1
pp_is,a∼est,une=1	pp_10_is,a∼est,une=1
pp_flash,of∼flash,de=1	pp_10_flash,of∼flash,de=1
...	...

HTMMs compute $P(z_n, \psi_n | d, w_{i=1}, .., w_N)$ for each sentence, where z_n is the topic of sentence n, d is the document and w_i are words in sentence n. ψ_n determines the topic transition between words and can be non-zero only at sentence boundaries. When $\psi_n = 0$, the topic is identical to the previous topic, when $\psi_n = 1$, a new topic is drawn from a distribution θ_d. Once the sentence topic has been selected, all w_i are generated according to a multinomial distribution with topic-specific parameters. In order to assign topics to sentences in our training data, we derive a sentence topic distribution

$$\begin{aligned} P(topic|sentence) &= P(z_n|d, w_{i=1}, .., w_N) \\ &= P(z_n, \psi_n = 0|d, w_{i=1}, .., w_N) \\ &+ P(z_n, \psi_n = 1|d, w_{i=1}, .., w_N) \end{aligned} \quad (1)$$

We noticed that the distributions $P(topic|sentence)$ were quite peaked in most cases and therefore we tried to use a more compact representation. First, we selected the most likely topic according to the topic distribution and treated this as ground truth, ignoring all other possible topics. Alternatively, we selected the two most likely topics along with their probabilities, ignoring the second most likely topics with a probability lower than 30%. The topic probabilies were then used instead of the binary feature values in order to integrate the confidence of the topic model in its assigments. Experimental results were slightly better for the first representation without probabilities and therefore we chose this simpler presentation in all reported experiments.

In order to improve the quality of the topic models, we used stop word lists and lists of salient TED talk terms to clean the in-domain data before training the topic models.

Table 2: *Sample English and German HTMM topics and their interpretation in quotes.*

"cancer"	"ocean"	"body"	"universe"
cancer	water	brain	universe
cells	ice	human	space
body	surface	neurons	Earth
heart	Earth	system	light
blood	Mars	mind	stars
Krebs	Wasser	DNA	Erde
Patienten	Meer	Leben	Universum
Gehirn	Menschen	Licht	Planeten
Zellen	Ozean	Bakterien	Leben
Körper	Tiere	Menschen	Sonne

All TED talks come with a small set of keywords (∼300 in total) describing the content of the talk. The idea was to use the information contained in these keywords to select salient terms that frequently cooccur with the keywords. We first computed tf-idf for all words in each talk, normalised by the number of words in the talk. We then summed up the normalised tf-idf counts for each keyword, i.e. the counts of words in all documents associated with a particular keyword, and selected the top 100 terms for each keyword. This yielded ∼10500 terms for English and ∼11700 terms for German.

In cases where this filtering yielded empty sentences in the in-domain data (sentences with no salient terms), the topic information was replaced by "unk". We ran the topic training for 100 iterations and trained 30 topics over training, development and test sets. We modified the Moses decoder to accept topic information as XML mark-up and annotated all data with sentence-wise topics (and optionally the respective probabilities). Table 2 gives some examples of topics and their 5 most frequent terms for English and German as a source language, as we use topic triggers associated with the source sentence for our sparse features. The topic models represent topics as integers but here we have added labels to indicate the nature of the topics and we selected topics that map across the two languages. In general, the topics do not neccessarily map to equivalent topics in another language.

Table 3 shows a sequence of training sentences and their most likely topic (as well as the second most likely topic if applicable). We can see that for some of the sentences, the model assigns what we have labelled the "universe" topic with high probability while for others it is less certain or makes a transition to the "ocean" topic.

2.4. Jackknife setup

Training sparse features always involves a risk of overfitting on the tuning set, especially with highly lexicalized features that might occur only once in the tuning set. Therefore, training sparse features on the entire training set used to estimate the phrase table is expected to be more reliable. For discriminative training methods this means that the training set needs to be translated in order to infer feature values and compute BLEU scores. However, translating the same data that was used to train the translation system would obviously cause overfitting as well, thus the system needs to be adjusted to prevent this. In order to translate the whole training data without bias, we apply the jackknife method to split up the training data into n=10 folds. We create n subsets of the training data containing n-1 folds and leaving out one fold at a time. These subsets serve as training data for n systems that can be used to translate the respective left-out fold.

Table 3: *Topic assignment to training sentences with topic probabilities in brackets.*

"universe" (0.41)	"And physicists came and started using it sometime in the 1980s."
"universe" (0.47)	"And the miners in the early part of the last century worked, literally, in candlelight."
"ocean" (0.71)	"And today, you would see this inside the mine, half a mile underground."
"ocean"/"universe" (0.51/0.49)	"This is one of the largest underground labs in the world."
"universe" (0.99)	"And, among other things, they're looking for dark matter."
"universe" (1.00)	"There is another way to search for dark matter, which is indirectly."
"universe" (1.00)	"If dark matter exists in our universe, in our galaxy, then these particles should be smashing together..."

To use the jackknife systems for MIRA training, we modified the algorithm to accept n sets of decoder configuration files, input files and reference files. Instead of running n instances of the same translation system in parallel, we run n jackknife systems in parallel and average their weight vectors several times per epoch.

When applying the jackknife method to the TED in-domain data, we noticed a problem with this approach. Usually it would be good practice to create folds in a way that the resulting subsets of training data are as uniform as possible in terms of vocabulary to minimize the performance hit caused by the missing fold. However, the vocabulary of the TED data turned out to be quite repetitive within sentences belonging to the same talk. Thus, splitting up the data uniformly had the effect that each of the n systems had a certain amount of phrasal overlap with its left-out fold. This resulted in a preference for longer phrases, overly long translations on the test set and decreasing performance during MIRA training.

We were able to overcome the overfitting effect of linewise data splits by splitting the data in a roughly talk-wise fashion instead. That is, the first x =corpus size/n lines were assigned to fold 1, the following x lines to fold 2 and so on. This way the folds were still the same size, but the training

data was much less likely to overlap with the left-out fold. The results on a held-out set during MIRA training (in particular the length penalty and overall length ratio) showed that this helped to prevent overfitting on the left-out fold.

3. Integrating features into mixed-domain models (retuning)

Tuning sparse features on top of large translation models can be time and memory-consuming. Especially the jackknife approach would cause immense overhead to tune with the mixed-domain data because we would need to train *n* different phrase tables that all include most of the in-domain data and all of the out-of-domain data[1]. Therefore, we wanted to investigate whether there is an alternative way of tuning our features on all of the in-domain data while also making use of the out-of-domain data. Tuning with the in-domain models allows for more flexibility in the training setup because the data set is relatively small. Since our goal is to translate documents of the TED talks domain, we assume that tuning sparse features only on the TED domain should provide the model with enough information to select the appropriate vocabulary. Hence we propose to port the tuned features from the in-domain models to the mixed-domain models. The advantage of this method is that features can be tuned on all the in-domain training data (jackknife) or in other ways that are feasible on a smaller in-domain model but might not scale well on a large mixed-domain model.

However, porting tuned feature weights from one model to another is not straightforward because the scaling of the core features is likely to be different. Therefore, to bring the sparse feature weights on the right scale to integrate them into the mixed-domain model, we perform a retuning step with MIRA. We take the sparse features tuned with the jackknife method and combine them into one aggregated meta-feature with a single weight. During decoding, the weight of the meta-feature is applied to all sparse features belonging to the same class (word pair or phrase pair features). In the retuning step, the core weights of the mixed-domain model are tuned together with the meta-feature weight.

An overview of our tuning schemes is given in figure 1. The training step denotes the entire training pipeline yielding the baseline models. Direct tuning refers to tuning with MIRA on a small development set and applies to both kinds of baseline models, while jackknife tuning only applies to in-domain models and retuning only to mixed-domain models.

4. Experiments

We evaluate our training schemes on English-French (en-fr) and German-English (de-en) translation systems trained on the data sets as advised for the IWSLT2012 TED task. As in-domain data we used the TED talks from the WIT[3] website[2] [2]. As out-of-domain data we used the Europarl, News Commentary and MultiUN [8] corpora and for en-fr also the 10^9 corpus taken from the WMT2012 release. An overview of all training data as well as development and test data is given in table 4 (sentence counts).

With this data we trained in-domain and mixed-domain baselines for both language pairs. For the mixed-domain baselines (trained on data from all domains), we used simple concatenations of all parallel training data, but trained separate language models for each domain and linearly interpolated them on the development set. All systems are phrase-based systems trained with the Moses toolkit [13]. Compound splitting and syntactic pre-reordering was applied to all German data. As optimizers we used MERT as implemented in the current version of Moses and a modified version of the MIRA implementation in Moses as described in section 2.1. We provide baseline results for tuning with both MERT and MIRA for comparison, though our model extensions are evaluated with respect to the MIRA baselines. Reported BLEU scores were computed using the mteval-v11b.pl script.

Figure 1: *In-domain (IN) and mixed-domain (IN+OUT) models with three tuning schemes for tuning sparse feature weights: direct tuning, jackknife tuning and retuning.*

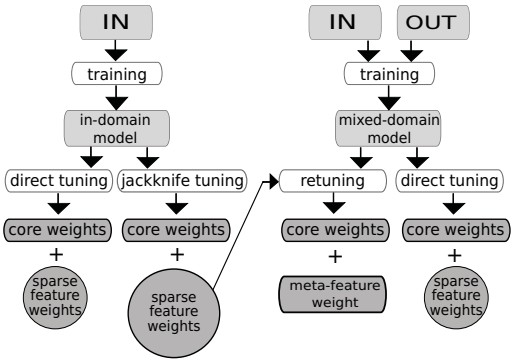


Table 4: *Sentence counts of in-domain (TED talks) and out-of-domain training data used in our systems.*

	en-fr	de-en
TED talks	140K (1029 talks)	130K (976 talks)
Europarl v7	2M	1.9M
News Commentary v7	137K	159K
MultiUN	12.9M	161K
10^9 corpus	22.5M	n/a
total	35.9M	2.3M
TED talks (monoling.)	143K	142K
dev2010	934 (8 talks)	900 (8 talks)
test2010.part1	898 (5 talks)	665 (5 talks)
test2010.part2	766 (6 talks)	900 (6 talks)

[1]Training the mixed-domain system for the en-fr language pair took more than a week.

[2]https://wit3.fbk.eu/mt.php?release=2012-03

All experiments except the jackknife experiments used the TED dev2010 set as development set (dev). The TED test2010 set was split into two parts, test2010.part1 and test2010.part2. For the in-domain experiments, one part was used to select the best weights found during MIRA training and the other part was used for evaluation, respectively. We refer to these sets as test1 and test2 to indicate which of the two parts was used as the test set. We note that test1 and test2 yield quite different BLEU scores for the baseline models. However, table 5 shows that the relative improvements achieved with MIRA are roughly proportional and thus we will report results on just one of the two sets for experiments on the mixed-domain baselines.

All MIRA experiments were initialized with the tuned weights of the MERT baselines. MIRA experiments on the dev set were run for 20 epochs, retuning experiments for 10 epochs and jackknife experiments on the entire training set for 2 epochs.

4.1. Results

We are evaluating the impact of our sparse features on the in-domain and mixed-domain systems. Tables 5 and 6 show the results on the in-domain system with BLEU scores reported on both parts of the test2010 set, using the respective other part as devtest set. Improvements over the MIRA baseline are marked in bold print and the relative changes are indicated in brackets. First we note that MIRA training improves the MERT baseline performance for the en-fr system by 0.8 BLEU on both test sets, but decreases performance for the de-en system by 0.3 BLEU. We believe that this divergence has to do with the changes in length ratio after MIRA training, as shown in table 7. For en-fr, translations get longer during MIRA training while for de-en they get shorter, incurring an increased brevity penalty according to the BLEU score.

Since MIRA has quite a different impact on the translation performance with the core features (translation model, reordering model, language model, word penalty, phrase penalty), we focus on the impact of sparse features with respect to the MIRA baselines. For en-fr, we observe that all sparse feature setups beat the MERT baseline and most of them beat the MIRA baseline. For the MIRA experiments on the dev set we notice that phrase pair features seem to perform better than word pair features on both test sets and sparse features with topic triggers seem to do better than sparse features without topic information. The results of the MIRA experiments using the jackknife method are in almost all cases better than the results trained on the small dev set. We get an increase of up to 1.3/0.2 BLEU (en-fr/de-en) over the MERT baseline and up to 0.5/0.7 BLEU (en-fr/de-en) over the MIRA baselines. This shows that the jackknife method is better suited to train sparse features than training on a small dev set. We still observe slightly better results for phrase pair features than for word pair features with the en-fr models, even though this observation is less conclusive than on the dev data.

Table 5: *In-domain baselines (IN) and results for sparse feature training on en-fr in-domain model, training on a development set (dev) and on all training data (jackknife).*

en-fr	BLEU(test1)	BLEU(test2)
MERT(dev) IN	*28.6*	*30.9*
MIRA(dev) IN	29.4	31.7
MIRA(dev)		
+ wp	29.2 (-0.2)	31.6 (-0.1)
+ wp + topics	**29.5** (+0.1)	**31.8** (+0.1)
+ pp	**29.6** (+0.2)	31.7 (+0.0)
+ pp + topics	**29.6** (+0.2)	**31.9** (+0.2)
MIRA(jackknife)		
+ wp	**29.7** (+0.3)	**32.2** (+0.5)
+ wp + topics	**29.5** (+0.1)	**32.1** (+0.4)
+ pp	**29.9** (+0.5)	**32.2** (+0.5)
+ pp + topics	**29.6** (+0.2)	**32.0** (+0.4)

Table 6: *In-domain baselines (IN) and results for sparse feature training on de-en in-domain model, training on a development set (dev) and on all training data (jackknife).*

de-en	BLEU(test1)	BLEU(test2)
MERT(dev) IN	*26.6*	*29.9*
MIRA(dev) IN	26.3	29.6
MIRA(dev)		
+ wp	**26.7** (+0.4)	**29.8** (+0.2)
+ wp + topics	**26.6** (+0.3)	**29.7** (+0.1)
+ pp	**26.5** (+0.2)	**29.7** (+0.1)
+ pp + topics	**26.4** (+0.1)	**29.8** (+0.2)
MIRA(jackknife)		
+ wp	**27.0** (+0.7)	**30.1** (+0.5)
+ wp + topics	**26.4** (+0.1)	**29.7** (+0.1)
+ pp	**26.8** (+0.5)	**30.0** (+0.4)
+ pp + topics	**26.4** (+0.1)	**29.8** (+0.2)

Tables 8 and 9 show results on the mixed-domain models, where we observe a similar divergence in performance between the MERT and MIRA baselines as on the in-domain models: a plus of 1.1 BLEU for en-fr and a minus of 0.4 BLEU for de-en. The first block of results refers to MIRA training on the dev2010 set as for the in-domain models (direct tuning), while the second block results from the retuning setup described in section 3 (retuning). The direct approach gains up to 0.5 BLEU for en-fr and up to 0.1 BLEU for de-en over the MIRA baselines, retuning with MIRA and jackknife features gains up to 0.5 BLEU for en-fr and up to 0.4 BLEU for de-en over the MIRA baselines. This is another indication that sparse features trained with the jackknife method can leverage information from the in-domain training data to help the model select appropriate words and phrases for the target domain. In some cases we can observe that topic

Table 7: *Changes to the length ratio (hypotheses/reference, in brackets) between MERT and MIRA tuning, indicated by (+) and (-).*

		BLEU(test1)	BLEU(test2)
en-fr	MERT(dev) IN	28.6 (0.969)	30.9 (0.963)
	MIRA(dev) IN	29.4 (0.987) (+)	31.7 (0.982) (+)
de-en	MERT(dev) IN	26.6 (0.987)	29.9 (1.001)
	MIRA(dev) IN	26.3 (0.955) (-)	29.6 (0.969) (-)

Table 8: *Mixed-domain baselines (IN+OUT) and results for sparse feature training on en-fr mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.*

en-fr	BLEU(test1)
MERT(dev) IN+OUT	*30.0*
MIRA(dev) IN+OUT	31.1
MIRA(dev), direct tuning	
+ wp	**31.6** (+0.5)
+ wp + topics	**31.4** (+0.3)
+ pp	**31.4** (+0.3)
+ pp + topics	**31.5** (+0.4)
MIRA(dev), retuning	
+ wp	**31.6** (+0.5)
+ wp + topics	31.1 (+0.0)
+ pp	**31.5** (+0.4)
+ pp + topics	**31.3** (+0.2)

features improve over simple features, even though they perform weaker in more of the cases. We suspect that sparsity issues need to be addressed to benefit more from these features. In general, the results show that features trained only on in-domain models can help to improve performance of much larger mixed-domain models. While for the in-domain models the results on both language pairs are similar w.r.t. the MIRA baselines, the results on mixed-domain models are clearly better for en-fr which can be considered an easier language pair for translation than de-en.

The feature sets ranged in size between around 5K-15K when training on a dev set and 60K-600K when training on all training data, depending on the particular feature type.

4.2. Topic features

For the en-fr in-domain systems trained on dev data, we see an improvement of topic features over simple sparse features. That these effects are not stronger might be due to the quite diverging distributions of topics across dev, devtest and test sets (see figure 2[3]). For example, the "universe" topic (topic 29) appears quite frequently in the training and dev data, but only twice in test2 and never in test1. For future experiments with sentence-level topic features it should be ensured that

[3]Training data counts were between 2252 and 7170 sentences per topic.

Table 9: *Mixed-domain baselines (IN+OUT) and results for sparse feature training on de-en mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.*

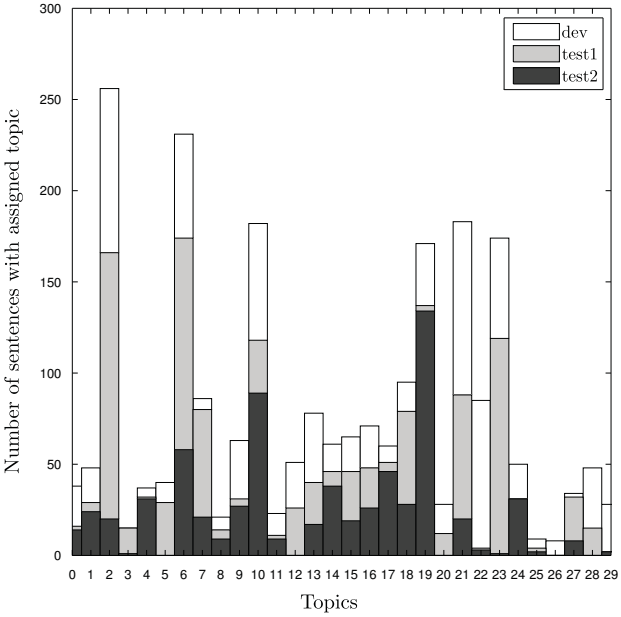
de-en	BLEU(test1)
MERT(dev) IN+OUT	*27.2*
MIRA(dev) IN+OUT	26.8
MIRA(dev), direct tuning	
+ wp	**26.9** (+0.1)
+ wp + topics	**26.9** (+0.1)
+ pp	**26.9** (+0.1)
+ pp + topics	26.7 (-0.1)
MIRA(dev), retuning	
+ wp	**27.1** (+0.3)
+ wp + topics	**27.2** (+0.4)
+ pp	**27.0** (+0.2)
+ pp + topics	**27.0** (+0.2)

topics are distributed more evenly across development sets.

Lexicalised features with topic triggers are even sparser than simple lexicalised features and therefore we would expect that they benefit particularly from jackknife training. However, our current results show the opposite tendency in that topic features seem to do worse than simple features under the jackknife setup. Table 10 gives an example of word pair features trained with the jackknife method, with and without topic information. It shows the features with the largest positive/negative weights (those with the highest discriminative power learned by the model) for translating the English source word "matter". Both models have learned that "matière" is the most appropriate French translation for the English word "matter". Both models penalize some translations of the other word sense like the French word "important". However, the model without topic information considers "importe" an almost equally likely translation, while the model with topic information penalizes all translations that do not preserve the physical word sense (as in "dark matter"). As mentioned above, the "universe" topic did not appear at all in test1, so the impact of features related to this topic has not been measured in the evaluation.

Table 11 shows jackknife-trained features for the source word "language". While with simple word pair features the most likely translation is "langage" (mode of speaking), the topic features express translation preferences according to the source topic. For example, given the "science" topic, the most likely translation is "langage", but given the "school" topic, the most likely translation is "langue". However, in table 1 we see that the input sentence is labelled with topic 10 ("science") but "language" is translated to "langue" in the reference translation. Thus, given the topic labelling the exptected translation with topic features would not match the reference translation, which is something that should be taken into account.

Figure 2: *Distribution of topics in dev, test1, test2.*



5. Related work

The domain adaptation literature can be broadly grouped into approaches adapting the language model and approaches adapting the translation model. Among the latter there has been work on mixture modeling of domain-specific phrase tables [9] and discriminative instance weighting [14] [10]. In similar spirit, [1] introduced a corpus-filtering technique that computes a bilingual cross-entropy difference to determine how similar a sentence pair is to an in-domain corpus and how dissimilar from a general-domain corpus. There has also been previous work on translation model adaptation using topics models. [19] employ HTMMs to train source-side topic models from monolingual in-domain data and the source side of parallel out-of-domain data. Phrase pairs are conditioned on in-domain topics via a mapping from in-domain to out-of-domain topics. Our approach is different in that we use parallel in-domain data and therefore do not need a mapping step. [7] extend previous work by [4] on lexical weighting conditioned on data provenance. They enhance lexical weighting features with topic model information to train separate word translation tables for every domain which can then be used to bias phrase selection based on source topics.

MIRA has been proposed for tuning machine translation systems with large features sets, for example by [20] and [3]. Recent work that compares tuning on a small development set versus tuning on the entire training data has been presented in [18]. The idea of using source triggers to condition word translation is somewhat related to the trigger-based lexicon models of [15], though they use context words as additional triggers and train their features with the EM algorithm.

Table 10: *Examples of en-fr jackknife-trained word pair features, with and without topic information (topic 29: "universe").*

sparse feature	feature weight
wp_matter∼matière	0.00170
wp_matter∼importe	0.00107
wp_matter∼important	-0.00037
wp_matter∼comptent	-0.00188
wp_29_matter∼matière	0.00431
wp_29_matter∼importent	-1.42913e-05
wp_29_matter∼importe	-0.00134
wp_29_matter∼important	-0.00172

Table 11: *Examples of en-fr jackknife-trained word pair features, with and without topic information (topic 10: "science", topic 27: "school").*

sparse feature	feature weight
wt_language∼langage	0.00444
wt_language∼langue	-0.00434
wt_10_language∼langage	0.01088
wt_10_language∼langue	-0.01071
wt_27_language∼langue	0.00792
wt_27_language∼langage	-0.00742

6. Conclusion

We presented a novel way of training lexicalised features for a domain adaptation setting by adding sparse word pair and phrase pair features to in-domain and mixed-domain models. In addition, we suggested a method of using topic information derived from HTMMs trained on the source language to condition the translation of words or phrases on the sentence topic. This was shown to yield improvements over simple sparse features on English-French in-domain models. We experimented with the jackknife method to use the entire in-domain data for feature training and showed BLEU score improvements for both language pairs. Finally, we introduced a retuning method for mixed-domain models that allows us to adapt features trained on the entire in-domain data to the mixed-domain models.

In the future, we would like to test our methods on hierarchical phrase-based or syntactic models. Other work in this field suggests that discriminative training yields larger gains with those types of models than with purely phrase-based models, so this would be an interesting comparison. We would also like to address the evaluation of topic features, which we believe requires a more controlled setting. Induced topics should be distributed more evenly across data sets and the quality of sentence topic labels should be taken into account.

7. References

[1] Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo In-Domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[2] Cettolo, M., Girardi, C., and Federico, M. (2012). Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of EAMT*, pages 261–268, Trento, Italy.

[3] Chiang, D., Knight, K., and Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of HLT: The 2009 Annual Conference of the NACL*, Stroudsburg, PA, USA. Association for Computational Linguistics.

[4] Chiang, D., DeNeefe, S., and Pust, M. (2011). Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, pages 455–460, Stroudsburg, PA, USA. Association for Computational Linguistics.

[5] Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. In *J. Machine Learning Research 13*, pages 1159–1187.

[6] Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(4-5):951–991.

[7] Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. Association for Computational Linguistics.

[8] Eisele, A. and Chen, Y. (2010). Multiun: A multilingual corpus from united nation documents. In *LREC'10*.

[9] Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.

[10] Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

[11] Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden Topic Markov Models. In *Journal of Machine Learnin Research, pp. 163-170*.

[12] Hasler, E., Haddow, B., and Koehn, P. (2011). Margin Infused Relaxed Algorithm for Moses. In *The Prague Bulletin of Mathematical Linguistics No. 96, 2011, pp. 69-78*, Prague.

[13] 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 *ACL 2007: proceedings of demo and poster sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

[14] Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

[15] Mauser, A., Hasan, S., and Ney, H. (2009). Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Conference on Empirical Methods in Natural Language Processing*, pages 210–217, Singapore.

[16] Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *ACL-2003: 41st Annual meeting of the ACL*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

[17] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL 02: Proceedings of the 40th Annual Meeting on ACL*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

[18] Simianer, P., Riezler, S., and Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the ACL*. Association for Computational Linguistics.

[19] Su, J., Wu, H., Wang, H., Chen, Y., Shi, X., Dong, H., and Liu, Q. (2012). Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. Association for Computational Linguistics.

[20] Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of EMNLP-CoNLL*, pages 764–773, Prague. Association for Computational Linguistics.

Interpolated Backoff for Factored Translation Models

Philipp Koehn
School of Informatics
University of Edinburgh
Scotland, United Kingdom
pkoehn@inf.ed.ac.uk

Barry Haddow
School of Informatics
University of Edinburgh
Scotland, United Kingdom
bhaddow@inf.ed.ac.uk

Abstract

We propose interpolated backoff methods to strike the balance between traditional surface form translation models and factored models that decompose translation into lemma and morphological feature mapping steps. We show that this approach improves translation quality by 0.5 BLEU (German–English) over phrase-based models, due to the better translation of rare nouns and adjectives.

1 Introduction

Morphologically rich languages pose a special challenge to statistical machine translation. One aspect of the problem is the generative process yielding many surface forms from a single lemma, causing sparse data problems in model estimation, affecting both the translation model and the language model. Another aspect is the prediction of the correct morphological features which may require larger syntactic or even semantic context to resolve.

Factored translation models (Koehn and Hoang, 2007) were proposed as a formalism to address these challenges. This modeling framework allows for arbitrary decomposition and enrichment of phrase-based translation models. For morphologically rich languages, one application of this framework is the decomposition of phrase translation into two translation steps, one for lemmata and one for morphological properties, and a generation step to produce the target surface form.

While such factored translation models increase robustness by basing statistics on the more frequent lemmata instead of the sparser surface forms, they do make strong independence assumptions. For frequent surface forms, for which we have rich statistics, there is no upside from the increased robustness, but there may be harm due to the independence assumptions.

Hence, we would like to balance traditional surface form translation models with factored decomposed models. We propose to apply methods common in language modeling, namely backoff and interpolated backoff. Our backoff models rely primarily on the richer but sparser surface translation model but back off to the decomposed model for unknown word forms. Interpolated backoff models combine surface and factored translation models, relying more heavily on the surface models for frequent words, and more heavily on the factored models for the rare words.

We show that using interpolated backoff improves translation quality, especially of rare nouns and adjectives.

2 Related Work

Factored translation models (Koehn and Hoang, 2007) were introduced to overcome data sparsity in morphologically rich languages. Positive results have been reported for languages such as Czech, Turkish, or German (Bojar and Kos, 2010; Yeniterzi and Oflazer, 2010; Koehn et al., 2010). The idea of pooling the evidence of morphologically related words is similar to the automatic clustering of phrases (Kuhn et al., 2010).

The popular Arabic–English language pair has received attention in the context of source language morphology reduction. Most work in this area involves splitting off affixes from complex Arabic words that translate into English words of their own (Sadat and Habash, 2006; Popović and Ney, 2004). A concentrated effort on reducing out-of-vocabulary words in Arabic is reported by Habash (2008), which includes the application of stemming, as we do here. However, in our work, we also address the translation of rare words and use a more complex factored decomposed model for the handling of unknown words. Backoff to stemmed models was explored by Yang and Kirchhoff (2006).

Corpus	Sentences	Words	
		English	German
Europarl	1,739,154	48,446,385	45,974,070
News Comm.	136,227	3,373,154	3,443,348
News Test 11	3,003	75,762	73,726

Table 1: Size of corpora used in experiments. Data from WMT 2011 shared tasks (Callison-Burch et al., 2011).

The idea of interpolated backoff stems from language modelling, where it is used in smoothing methods such as Witten-Bell (Witten and Bell, 1991) and Kneser-Ney (Kneser and Ney, 1995). See Chen and Goodman (1998) for an overview. Smoothing methods were previously used by Foster et al. (2006) to discount rare translations, but not in combination with backoff methods.

3 Anatomy of Lexical Sparsity

Before we dive into the details of our method, let us first gather some empirical insights into the problem we address.

Our work is motivated by overcoming lexical sparsity in corpora of morphologically rich languages. But how big is the portion of rare words in the test set and do we translate them significantly worse? We examined these questions on the German–English language pair, given the News Commentary and Europarl training corpora and the WMT 2011 test set (corpus sizes are given in Table 1). We trained a phrase-based translation model using Moses (Koehn et al., 2007) with mostly default parameters (for more details, please check the experimental section).

3.1 Computation of Source Word Translation Precision

The question, if a (potentially rare) input word has been translated correctly, does unfortunately not have a straight-forward answer: while *target* words can be compared against a reference translation, *source* words need to first tracked to their target word translations (if any), which then in turn can be compared against a reference.

We proceed as follows (see Figure 1). We record the word alignment within the phrase mappings, to closely track which input word was mapped to which output word. Note that the input word may have been dropped (has no word alignment in the phrase mapping), so we cannot proceed. We record those words as deleted and list them separately in our analysis.

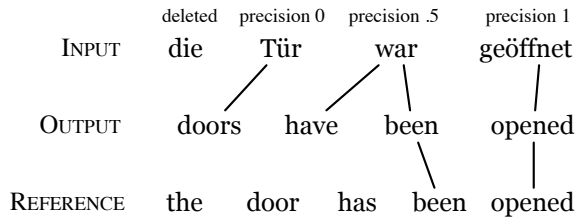


Figure 1: Computation of source word translation precision: Source words are traced to the target words that they are aligned to, which are in turn checked against a reference translation. *Tür* is aligned to *doors*, which is not in the reference, so precision is 0, *war* is aligned to two words, of which one is correct, and *geöffnet* aligned to a correct translation. Unaligned source words such as *die* are recorded as deleted.

We now have to determine if that output word is correct. To this end, we refer to the reference translation and check if the word can be found there. So, essentially, we compute the precision of a word translation.

There are a few fine points to observe: We may produce a word multiple times in our translation, but the reference may have it fewer times. In this case, we give only fractional credit. For instance, if the word occurs twice in our translation, but once in the reference, then producing the word counts only as 0.5 correct. We address many-to-many word alignments in a similar fashion.

3.2 Precision by Frequency

See Figure 2 for a graphical display of some of the findings of this study, primarily on translations using only the News Commentary corpus.

The first graph shows the precision of word translation (y-axis) with respect to the frequency of the word in the training corpus (x-axis). You will notice that we translate rare words only about 30% correctly, but about 50% of more frequent words. Very frequent words translate 70% correctly.

Words are categorized into bins based on the $\lceil \log_2(\text{count}) \rceil$. As additional information, we scaled the x-axis by the frequency of words of a given bin in the test set. You will notice that the bins have

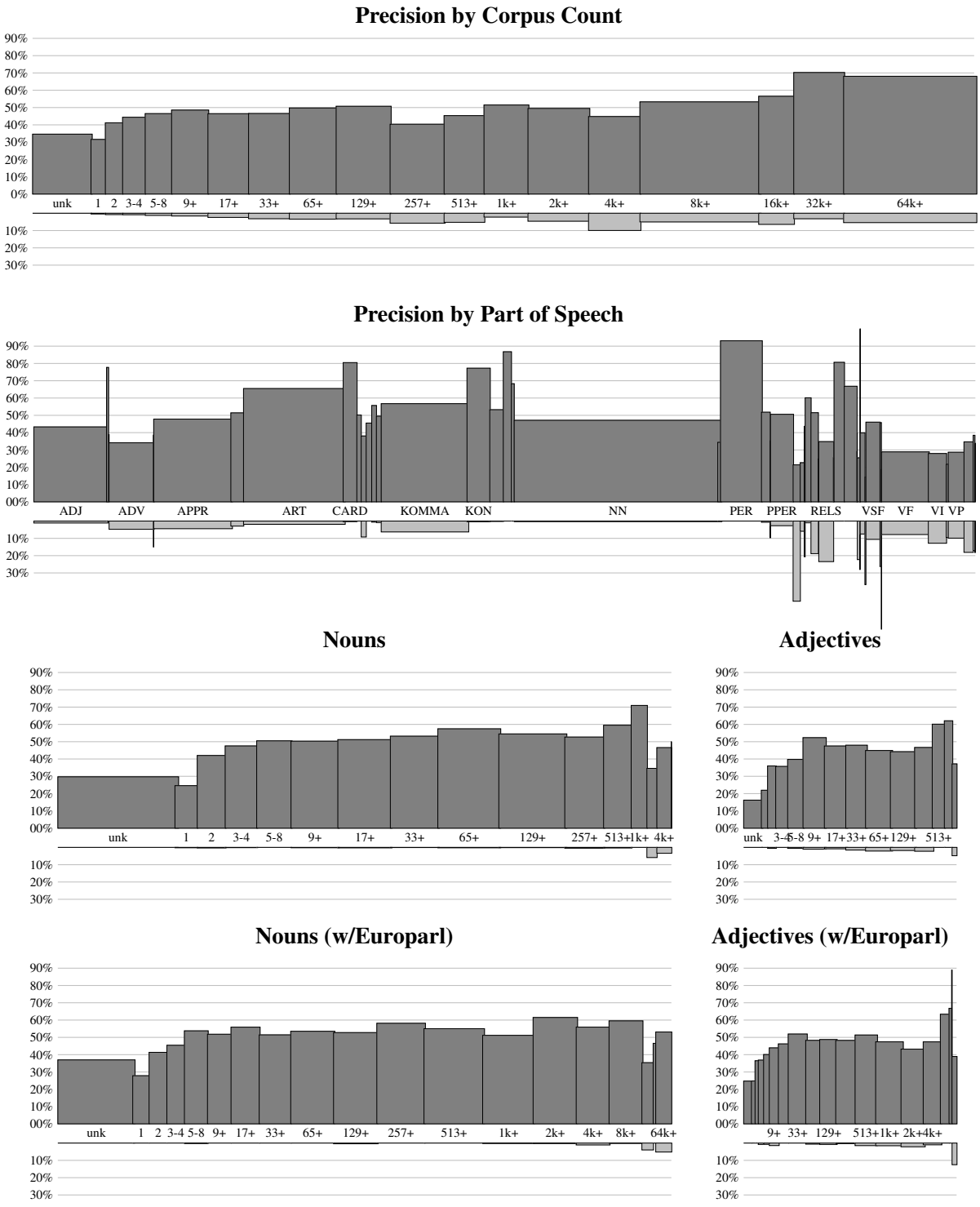


Figure 2: **Precision of the translation by type of source words.** The y-axis indicates precision for the upper part and the ratio of deleted words in the lower part of each graph. The x-axis scales each category (either words grouped by count in the training corpus, or by part-of-speech tag obtained with LoPar (Schmid and Schulte im Walde, 2000) using the Stuttgart Tag Set) by the number of occurrences of words in that category in the test set. Note the power law distribution of word frequencies: Even when increasing the training corpus by a factor of 15 when adding Europarl, there is still a large number of rare nouns and adjectives, which are less likely to be translated correctly.

roughly the size width: there are about as many words that occur 17-32 times in the training corpus, as there are words that occur 4097-8192 times. This is a nice reflection of Zipf's law. However, relatively few words in our test set occurred exactly once in the training corpus, while there is a significant number of unknown words.

3.3 Precision by Part-of-Speech

The relationship between frequency of a word in the training corpus and the precision of its translation is not clear cut. Part of the explanation is that some types of words are inherently easier to translate than others. The second graph breaks down translation precision by part of speech. Notable outliers are periods (PER) which we translate about 95% correctly, and verbs (V*) whose translation is very poor (about 30% correct). A good 10% of verbs are dropped during translation.

The main open class words are nouns and adjectives (verbs are also open class, but we found that there are only few rare verb forms). Since both nouns and adjectives are inflected in German, we want to pay special attention to them. The third row of graphs displays their precision by coverage.

Rare nouns and adjectives translate significantly worse than frequent ones: less than 25% of singletons are translated correctly vs. up to 60% of the very frequent ones. A good number of unknown nouns and adjectives are translated correctly (about 30%), since many of them are names (e.g., *Flickr, Piromalli, Mainzer*) which are just placed in the output unchanged.

About half of the nouns and adjectives in the test set occur less than 32 times in the training corpus. When we add the 15 times bigger Europarl corpus, the frequency of words increases, but not at the same rate as the corpus increase. There are still significant number of rare nouns left — roughly a third occur less than 32 times.

It is worthwhile to point out that nouns carry a substantial amount of meaning and their mistranslation is typically more serious than a dropped determiner or punctuation token. Translating them well is important.

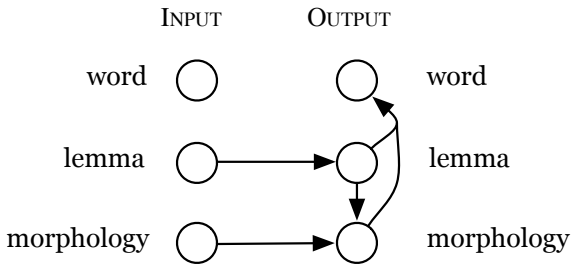


Figure 3: Factored translation model: Phrase translation is decomposed into a number of mapping steps.

4 Method

Our method involves a traditional phrase-based model (Koehn et al., 2003) and a factored translation model (Koehn and Hoang, 2007). The traditional phrase based model is estimated using statistics on phrase mappings found in an automatically word-aligned parallel corpus.

4.1 Decomposed Factored Model

The factored translation model decomposes the translation of a phrase into a number of mapping steps. See Figure 3 for an illustration. The decomposition involves two translation steps (between lemmata and between morphologically features) and two generation steps (from lemma to morphologically features and for the generation of the surface from both).

Formally, we introduce latent variables for the English lemma e_l and morphology e_m, in addition to the observed foreign morphological analysis f_s, f_l, f_m and the predicted English surface form e_s.

$$p(e_s|f_s, f_l, f_m) = \sum_{e_l, e_m} p(e_s, e_l, e_m|f_s, f_l, f_m) \quad (1)$$

However, we do not sum over all derivations, but limit ourselves to the best derivation.

$$p(e_s|f_s, f_l, f_m) \simeq \max_{e_l, e_m} p(e_s, e_l, e_m|f_s, f_l, f_m) \quad (2)$$

The fully-factored model is decomposed into three mapping steps using the chain rule.

$$\begin{aligned} p(e_s, e_l, e_m|f_s, f_l, f_m) = \\ p(e_m|f_s, f_l, f_m) \times \\ p(e_l|e_m, f_s, f_l, f_m) \times \\ p(e_s|e_l, e_m, f_s, f_l, f_m) \end{aligned} \quad (3)$$

A number of independence assumptions simplify the probability distributions for the mapping steps.

$$p(e_s,e_l,e_m|f_s,f_l,f_m) \simeq p(e_m|f_m)\ p(e_l|f_l)\ p(e_s|e_l,e_m) \quad (4)$$

Probability distributions for the mapping steps are estimated from a word-aligned parallel corpus. This data is processed so that each word is annotated with its lemma and morphological features (part-of-speech, case, count, gender, tense, etc.). As in traditional phrase-based models, translation steps are estimated from statistics of phrase mappings, but over the factor of interest.

The generation model $p(e_s|e_l,e_m)$ are estimated from a monolingual target-side corpus. These models are further decomposed to the word-level. For instance, for a two-word target side phrase, each word is generated independently from the predicted lemma and morphological features.

Note that we add a generation model $p(e_m|e_l)$ which is less mathematically motivated, but empirically effective. We discuss additional probability distributions towards the end of this section.

4.2 Backoff

The backoff model primarily relies on the phrase-based model. Only for unknown words and phrases, the secondary factored model is consulted for possible translations. We may limit the backoff to the secondary models to words, short phrases, or for phrases of any length.

Formally, we back off from a conditional probability distribution $p_1(e|f)$ to a secondary probability distribution $p_2(e|f)$ if there is no observed count of f in the training corpus for the earlier.

$$p_{\text{BO}}(e|f) = \begin{cases} p_1(e|f) & \text{if count}_1(f) > 0 \\ p_2(e|f) & \text{otherwise} \end{cases} \quad (5)$$

Note that we could create a backoff chain of more than two models, although we do not do so in this work. For instance, we may introduce a third model that relies on synonyms or paraphrasing to increase coverage.

This use of backoff is similar to its use in n-gram language models Chen and Goodman (1998); Stolke (2002). For unknown histories, these models back off to lower order n-gram models. We do not, however, mirror the behavior of backing off to lower order n-gram models for known histories but unknown predicted words. We will explore this idea in the next section.

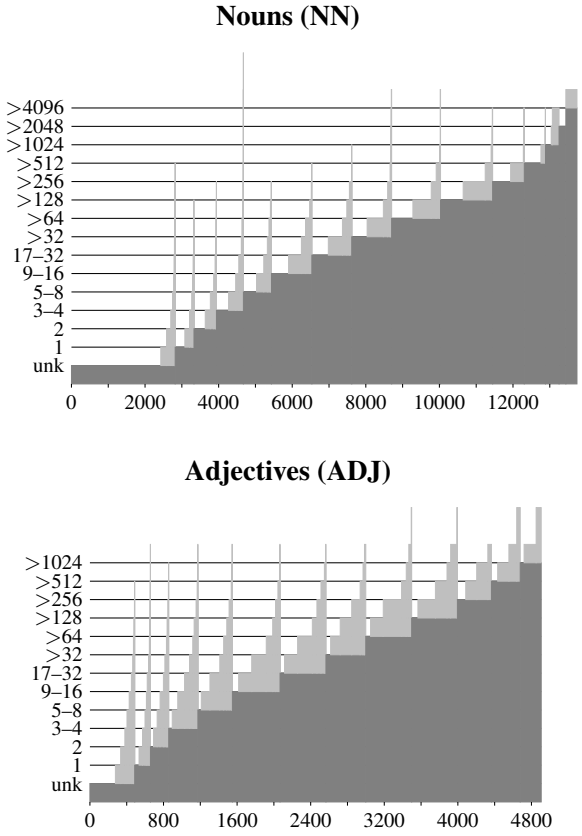


Figure 4: **Increased coverage for lemmata vs. annotated surface forms:** Given the corpus count of a word (dark gray), how much higher is the count for its lemma (light gray)? All counts are binned using $\log_2$. The x-axis is scaled according to the frequency of words of each count bin in the test set.

Figure 4 illustrates how the increase in corpus counts for lemmata opposed to annotated surface forms, indicating the potential for finding correct backoff translations.

4.3 Interpolated Backoff

While the backoff model will allow us to use the decomposed factored model for *unknown* surface forms, it does not change predictions for *rare* surface forms f — words that may have been only seen once or twice.

The idea of interpolated backoff is to subtract some of the probability mass from translations e in the primary distribution $p_1(e|f)$ and use it for additional (or identical) translations from the secondary distribution $p_2(e|f)$. We first convert $p_1(e|f)$ into a function $\alpha(e|f)$, and use the remaining probability mass for $p_2(e|f)$.

$$p_{\text{IBO}}(e|f) = \alpha(e|f) + \left(1 - \sum_e \alpha(e|f)\right) p_2(e|f) \quad (6)$$

We obtain $\alpha(e|f)$ by absolute discounting. Instead of estimating the translation probability mass from counts in the training corpus by maximum likelihood estimation

$$p_1(e|f) = \frac{\text{count}(e, f)}{\sum_e \text{count}(e, f)} \quad (7)$$

we subtract a fixed number D from each count when deriving probabilities for observed translations e

$$\alpha(e|f) = \frac{\text{count}(e, f) - D}{\sum_e \text{count}(e, f)} \quad (8)$$

4.4 Multiple Scoring Functions

Phrase-based models do not just use the direct phrase translation probabilities $p(e|f)$, but also their inverse $p(f|e)$ and bi-drectional lexical translation (IBM Model 1 or similar). In our experiments all these four scoring functions are used in the phrase-based model and in the translation steps of the decomposed factored model.

We compute a uniform discount factor for all four scoring functions from the count statistics for the direct translation probability distribution. This factor becomes apparent when reformulating the computation of $\alpha(e|f)$.

$$\alpha(e|f) = \frac{\text{count}(e, f) - D}{\text{count}(e, f)} p_1(e|f) \quad (9)$$

We apply the same factor to the other three scoring functions, for instance:

$$\alpha(f|e) = \frac{\text{count}(e, f) - D}{\text{count}(e, f)} p_1(f|e) \quad (10)$$

The factored translation model also consists of a number of scoring functions (four for each translation tables, one for each generation table). All these are used in the backoff model. For the interpolated backoff model, we need to combine the many scoring functions of the decomposed factored models into the four scoring functions of the translation model (phrase translation and lexical translation, in both directions).

We do so, scaling the four scoring functions of the lemma translation step $p(e_l|f_l)$ with

- direct morphology translation $p(e_m|f_m)$
- lemma to morphology generation $p(e_m|e_l)$
- surface form generation $p(e_s|e_m, e_l)$

Note that the three scaling probabilities are typically close to 1 for the most likely predictions. The surface generation probability is almost always 1.

See Figure 5 for an example of this process.

5 Experiments

We carry out all our experiments on the German–English language pair, relying on data made available for the 2011 Workshop for Statistical Machine Translation (Callison-Burch et al., 2011). Training data is from European Parliament proceedings and collected news commentaries. The test set consists of a collection of news stories. As is common for this language set, we perform compound splitting (Koehn and Knight, 2003) and syntactic pre-reordering (Collins et al., 2005).

We annotate input words and output words with all three factors (surface, lemma, morphology). This allows us to use 5-gram lemma and 7-gram morphology sequence models to support language modeling. The lexicalized reordering model is based on lemmata, so we can avoid inconsistencies between its use for translations from the joint and decomposed factored translation models. Word alignment is also performed on lemmata instead of surface forms. Phrase length is limited to four words, otherwise default Moses parameters are used. The fully-factored phrase-based model outperforms a pure surface form phrase-based model (+.30 BLEU).

The factored model has been outlined in Section 4.1. We used the following tools to generate the factors:

- English lemma: porter stemmer (Porter, 1980)
- English morphology (just POS): MXPOST (Ratnaparkhi, 1996)
- German lemma and morphology: LoPar (Schmid and Schulte im Walde, 2000)

Translations for morphological variants of *scheinheiliger* [ADJ.R; *scheinheilig*]

Surface	Translation	Count	$p_1(e\|f)$	$\alpha(e\|f)$
scheinheilig [ADJ.PRED; *scheinheilig*]	*hypocritical* [JJ; *hypocrit*]	5	1.00	0.90
scheinheilige [ADJ.E; *scheinheilig*]	*hypocrisy* [NN; *hypocrisi*] *of* [IN; *of*]	1	0.33	0.17
	hypocritical [JJ; *hypocrit*]	1	0.33	0.17
	hypocrisy [NN; *hypocrisi*]	1	0.33	0.17
scheinheiligen [ADJ.N; *scheinheilig*]	*hypocritical* [JJ; *hypocrit*]	1	0.50	0.25
	sanctimonious [JJ; *sanctimoni*]	1	0.50	0.25
scheinheiliger [ADJ.R; *scheinheilig*]	*of* [IN; *of*] *hypocrisy* [NN; *hypocrisi*]	1	1.00	0.50

Translations of lemma *scheinheilig*

Translation	Count	$p(e_l\|f_l)$
hypocrit	7	0.63
hypocrisi of	1	0.09
hypocrisi	1	0.09
sanctimoni	1	0.09
of hypocisi	1	0.09

Relevant translations of morphological tag ADJ.R

Translation	$p(e_m\|f_m)$
JJ	0.749
NN	0.042
IN NN	0.001
NN IN	0.005

Generation of English morphology given lemma

Lemma	Morphology	$p(e_m\|e_l)$
hypocrit	JJ	0.793
	NN	0.103
hypocrisi	JJ	0.018
	NN	0.891
sanctimoni	JJ	0.667
of	IN	0.999

Selected generated valid surface forms

Lemma	$p(e_l\|f_l)$	Morph.	$p(e_m\|f_m)$	$p(e_m\|e_l)$	Surface	$p_2(e\|f)$	$\alpha(e\|f)$	$p(e\|f)$
hypocrit	0.63	JJ	0.749	0.793	*hypocritical*	0.374	0.000	0.187
hypocrit	0.63	NN	0.042	0.103	*hypocrit*	0.003	0.000	0.002
hypocrisi	0.09	NN	0.042	0.891	*hypocrisy*	0.003	0.000	0.002
sanctimoni	0.09	JJ	0.749	0.667	*sanctimonious*	0.045	0.000	0.023
of hypocrisi	0.09	IN NN	0.001	0.999×0.891	*of hypocrisy*	0.000	0.500	0.500

Figure 5: **Example for interpolated backoff:** For the annotated surface form *scheinheiliger* [ADJ.R; *scheinheilig*], we discount the probability for the only existing translation (assuming absolute discounting of 0.5), and consult the decomposed factored model for additional translations. The highly likely translation *hypocritical* is added with probability 0.184, alongside other translations (slight simplified actual example from model).

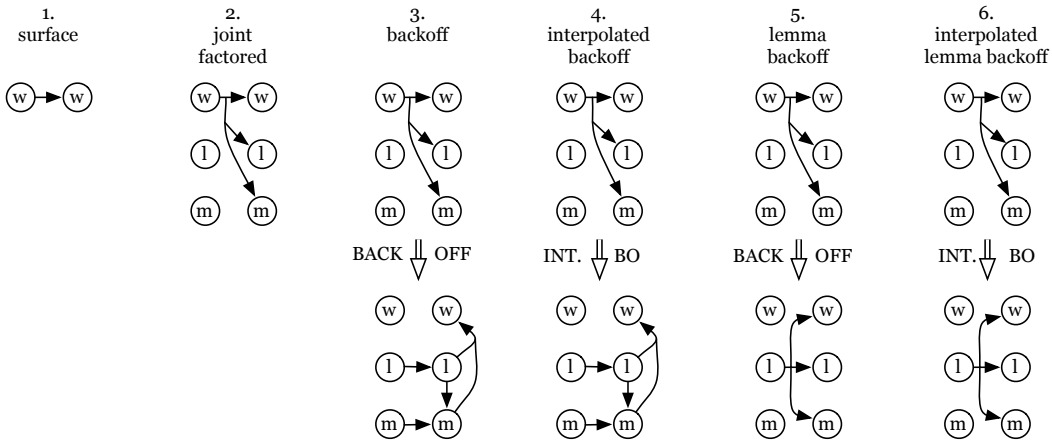


Figure 6: Six experimental configurations compared in Table 2.

When translating from a morphologically rich language, we would like to back off to lemma translation for unknown and rare input surface forms. Backing off only for unknown input words is the backoff method described in Section 4.2.

Interpolated backoff combines the phrase-based model with the decomposed factored model for rare input words and phrases (Section 4.3). For our experiments, we used a discount value of 0.5 and only performed interpolated backoff for input words that occurred at most 7 times. We also experimented with different discount values but did not achieve higher performance.

In Table 2, we report case-sensitive BLEU scores for the following models (illustrated in Figure 6):

1. a plain **surface** phrase-based models that uses only surface forms
2. a **joint factored** models that translates all factors (surface, lemma, morphology) in one translation step, employing additional n-gram models
3. a **backoff** model (Section 4.2) from the joint phrase-based model to the decomposed model (Section 4.1)
4. an **interpolated backoff** model, same as above, but with adjustments to rare word translations (Section 4.3)
5. a **lemma backoff** model from the joint phrase-based model to a model that maps from source lemmata into all target factors
6. an interpolated backoff version of above

Model	**NewsComm.**	**NC+Europl.**
1. surface	16.53	21.43
2. joint factored	16.83 (+.30)	21.54 (+.11)
3. backoff	16.96 (+.43)	21.63 (+.20)
4. int. backoff	17.03 (+.50)	21.65 (+.22)
5. lemma backoff	16.95 (+.42)	21.58 (+.15)
6. lemma int-back.	16.95 (+.42)	21.60 (+.17)
best single system at WMT2011		21.8

Table 2: Improvement (BLEU) in overall translation quality of the backoff methods for German–English.

For models trained only on the 3 million word News Commentary corpus, we see gains for both backoff (+0.43 BLEU) and interpolated backoff (+0.50 BLEU). For models that also included the Europarl corpus as training data (about 15 times bigger), we see gains each of the methods (+0.20 BLEU and +0.22 BLEU, respectively). Part of these gains stem from the original joint factored model, so the gains attributable to the backoff strategies are about half of the stated numbers.

Overall, the numbers are competitive with the state of the art – the best single system (Herrmann et al., 2011) at the WMT 2011 shared task scored 0.15 BLEU better (according to scores reported at http://matrix.statmt.org/) than our best system here.

For the large NC+Europarl training set, tuning with PRO (Hopkins and May, 2011) was run five times and the average of the test scores are reported (although results do often not differ by much more

Count (training)	News Commentary	
	Adjectives	**Nouns**
unk	27.2% (+11.0%)	31.1% (+1.4%)
1	27.5% (+6.6%)	28.0% (+4.0%)
2	37.8% (+5.9%)	43.5% (+2.8%)
3–4	36.6% (+2.6%)	49.1% (+0.7%)
5–8	37.8% (–0.3%)	51.3% (+0.5%)
	News Commentary + Europarl	
unk	29.2% (+4.5%)	37.5% (+0.5%)
1	27.8% (+3.0%)	31.0% (+3.2%)
2	39.2% (+2.7%)	43.2% (+1.9%)
3–4	41.1% (+4.3%)	46.7% (+1.3%)
5–8	45.4% (+5.3%)	53.7% (–0.1%)

Table 3: Improved precision of the translation of rare adjectives and nouns for the combined backoff methods.

than 0.01).

The lemma models are included to examine if our gains come from the fact that we are able to translate words whose lemma we have seen, or if there are any benefits to use the decomposed factored model. The results show that we do see higher gains with the decomposed factored model (+.08 and +.05 BLEU for the interpolated backoff model for the two corpora).

Our models do not back off (or compute interpolated backoff probabilities) for phrases longer than one word. We did not observe any gains from backing off for longer phrases, but incurred significant computational cost.

6 Analysis

Our methods target the translation of rare words, so we would only expect improvements in the translation of frequent words as knock-on effect. How much improvements do we see in the translation of rare words? Table 3 gives a summary.

We observe the biggest improvement for the translation of unknown adjectives in the News Commentary data set (+11.0%), we also see gains for singleton words (+3.0% to +6.6%) and twice-occurring words (+1.9% to +5.9%), and less pronounced gains for more frequent words. We see more gains for adjectives than nouns, since they have more morphological variants.

It is interesting to consider two examples that show the impact of the interpolated back-off model:

(1) The German word *Quadratmeter* (English *square meter*) was translated incorrectly by the simple backoff model, since the word occurred in the training corpus only twice, once with the correct and once with a wrong translation. The interpolated backoff model arrived at the correct translation since it benefitted from the additional three correct translation of morphological variants.

(2) However, the German word *Gewalten* was translated incorrectly into *violence* by the interpolated backoff model, while the simple backoff model arrived at the right translation *powers*. The word occurred only three times in the corpus with the acceptable translations *powers, forces,* and *branches*, but its singular form *Gewalt* is very frequent and almost always translates into *violence*.

These examples show the strengths and weaknesses of interpolated backoff. Considering the translations of morphological variants is generally helpful, except when these have different meaning, as it is sometimes the case with singular and plural nouns (an English example is *people* and *peoples*).

7 Conclusion

We introduced backoff methods for the better translation of rare words by combining surface word translation with translations obtained from a decomposed factored model. We showed gains in BLEU and improved translation accuracy for rare nouns and adjectives.

Acknowledgement The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 288769 (ACCEPT).

References

Bojar, O. and Kos, K. (2010). 2010 failures in english-czech phrase-based mt. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Chen, S. F. and Goodman, J. (1998). An emprirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

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.

Foster, G., Kuhn, R., and Johnson, H. (2006). Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia.

Habash, N. (2008). Four techniques for online handling of out-of-vocabulary words in arabic-english statistical machine translation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 57–60, Columbus, Ohio.

Herrmann, T., Mediani, M., Niehues, J., and Waibel, A. (2011). The karlsruhe institute of technology translation systems for the wmt 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 379–385, Edinburgh, Scotland. Association for Computational Linguistics.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Accoustics, Speech and Signal Processing*, volume 1.

Koehn, P., Haddow, B., Williams, P., and Hoang, H. (2010). More linguistic annotation for statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 121–126, Uppsala, Sweden.

Koehn, P. and Hoang, H. (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*.

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)*.

Kuhn, R., Chen, B., Foster, G., and Stratford, E. (2010). Phrase clustering for smoothing tm probabilities - or, how to extract paraphrases from phrase tables. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 608–616, Beijing, China.

Popović, M. and Ney, H. (2004). Towards the use of word stems and suffixes for statistical machine translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1585–1588.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 3(14):130–137.

Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*.

Sadat, F. and Habash, N. (2006). Combination of Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

Schmid, H. and Schulte im Walde, S. (2000). Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

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

Witten, I. H. and Bell, T. C. (1991). The zero-frequency problem: estimating the probabilities of novelevents in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094.

Yang, M. and Kirchhoff, K. (2006). Phrase-based backoff models for machine translation of highly inflected languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.

Yeniterzi, R. and Oflazer, K. (2010). Syntax-to-morphology mapping in factored phrase-based statistical machine translation from english to turkish. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 454–464, Uppsala, Sweden.

Towards Effective Use of Training Data in Statistical Machine Translation

Philipp Koehn and Barry Haddow
University of Edinburgh
Edinburgh, United Kingdom
{pkoehn,bhaddow}@inf.ed.ac.uk

Abstract

We report on findings of exploiting large data sets for translation modeling, language modeling and tuning for the development of competitive machine translation systems for eight language pairs.

1 Introduction

We report on experiments carried out for the development of competitive systems on the datasets of the 2012 Workshop on Statistical Machine Translation. Our main focus was directed on the effective use of all the available training data during training of translation and language models and tuning.

We use the open source machine translation system Moses (Koehn et al., 2007) and other standard open source tools, hence all our experiments are straightforwardly replicable[1].

Compared to all single system submissions by participants of the workshop we achieved the best BLEU scores for four language pairs (es-en, en-es, cs-en, en-cs), the 2^{nd} best results for two language pairs (fr-en, de-en), as well as a 3^{rd} place (en-de) and a 5^{th} place (en-fr) for the remaining pairs. We improved upon this in the post-evaluation period for some of the language pairs by more systematically applying our methods.

During the development of our system, we saw most gains from using large corpora for translation model training, especially when using subsampling techniques for out-of-domain sets, using large corpora for language model training, and larger tuning sets. We also observed mixed results with alternative tuning methods. We also experimented with hierarchical models and semi-supervised training, but did not achieve any improvements.

[1]Configuration files and instructions are available at http://www.statmt.org/wmt12/uedin/.

LP	Baseline	+UN
fr-en	28.2	28.4 (+.2)
es-en	29.1	28.9 (–.2)
en-fr	28.8	28.7 (–.1)
en-es	31.0	30.9 (–.1)
LP	**Baseline**	**+GigaFrEn**
fr-en	28.7	29.1 (+.4)
en-fr	29.3	30.3 (+1.0)

Table 1: Gains from larger translation models: UN (about 300 million English words), GigaFrEn (about 550 million English words).

We report all results in case-sensitive BLEU (mteval13a) on the newstest2011 test set (Callison-Burch et al., 2011). Please also note that baseline scores vary throughout the paper, since different methods were investigated at different time points.

2 Better Translation Models

2.1 Using Large Training Sets

The WMT evaluation campaign works with the largest training sets in the field. Our French-English systems are trained on a parallel corpus with 1,072 million French and 934 million English words. Training a system on this amount of data takes about two weeks.

The basic data sets for the language pairs are the Europarl and NewsCommentary corpora consist of about 50 million words and 3 million words, respectively. These corpora are quite close to the target domain of news reports, and give quite good results. Table 1 shows the gains from using the much larger UN (about 300 million words) and GigaFrEn corpora (about 550 million words).

From these results, it is not clear if the UN is helpful, but the GigaFrEn corpus gives large gains (+0.4 BLEU and +1.0 BLEU).

LP	Base-line	Model 1				Moore-Lewis			
		Before		After		Before		After	
		10%	50%	10%	50%	10%	50%	10%	50%
fr-en	29.3	28.5(–.8)	29.1(–.2)	28.6(–.7)	28.9(–.4)	29.1(–.2)	**29.6(+.3)**	29.1(–.2)	29.4(+.1)
en-fr	30.1	29.1(–1.0)	30.1(±.0)	29.3(–.8)	29.8(–.3)	29.9(–.2)	**30.2(+.1)**	29.9(–.2)	30.1(±.0)
es-en	29.0	28.9(–.1)	29.0(±.0)	29.0(±.0)	29.0(±.0)	29.0(±.0)	29.1(+.1)	**29.4(+.4)**	29.2(+.2)
en-es	30.9	30.9(±.0)	31.0(+.1)	30.8(–.1)	30.7(–.2)	31.4(+.5)	**31.5(+.6)**	**31.5(+.6)**	31.3(+.4)

Table 2: Subsampling UN and GigaFrEn corpora using Model 1 and Moore-Lewis filtering, before and after word alignment

2.2 Subsampling

We experimented with two different types of subsampling techniques – Model 1, similar to that used by Schwenk et al. (2011), and modified Moore-Lewis (Axelrod et al., 2011) – for the language pairs es-en, en-es, fr-en and en-fr. In each case the idea was to include the NewsCommentary and Europarl corpora in their entirety, and to score the sentences in the remaining corpora (the selection corpus) using one of the two measures, adding either the top 10% or top 50% of the selection corpus to the training data.

For Model 1 filtering, we trained IBM Model 1 on Europarl and NewsCommentary concatenated, in both directions, and scored the sentences in the selection corpus using the length-normalised sum of the IBM Model scores. For the modified Moore-Lewis filtering, we trained two 5-gram language models for source and target, the first on 5M sentences from the news2011 monolingual data, and the second on 5M words from the selection corpus, using the same vocabulary. The modified Moore-Lewis score for a sentence is the sum of the source and target's perplexity difference for the two language models.

For the Spanish experiments, the selection corpus was the UN data, whilst for the French experiments it was the UN data and the GigaFrEn data, concatenated and with duplicates removed.

The results of the subsampling are shown in Table 2, where the BLEU scores are averaged over 2 tuning runs. The conclusion was that modified Moore-Lewis subsampling was effective (and was used in our final submissions), but Model 1 sampling made no difference for the Spanish systems, and was harmful for the French systems.

3 Better Language Models

In previous years, we were not able to make use of the monolingual LDC Gigaword corpora due to lack of sufficiently powerful computing resources. These corpora exist for English (4.3 billion words), Spanish (1.1 billion words), and French (0.8 billion words). With the acquisition of large memory machines[2], we were now able to train language models on this data. Use of these large language models during decoding is aided by more efficient storage and inference (Heafield, 2011).

Still, even with that much RAM it is not possible to train a language model with SRILM (Stolke, 2002) in one pass. Hence, we broke up the training corpus by source (*New York Times*, *Washington Post*, ...) and trained separate language model for each. The largest individual corpus was the English *New York Times* portion which consists of 1.5 billion words and took close to 100GB of RAM. We also trained individual language models for each year of WMT12's monolingual corpus.

We interpolated the language models using the SRILM toolkit. The toolkit has a limit of 10 language models to be merged at once, so we had to interpolate sub-groups of some of the language models (the WMT12 monolingual news models) first. It is not clear if this is harmful, but building separate language model for each source and year and interpolate those many more models did hurt significantly.

Table 3 shows that we gain around half a BLEU point into Spanish and French, as well as German–English, and around one and a half BLEU points for the other language pairs into English.

[2]Dell Poweredge R710, equipped with two 6-core Intel Xeon X5660 CPUs running at 2.8GHz, with each core able to run two threads (24 threads total), six 3TB disks and 144GB RAM, and cost £6000.

LP	Baseline	+LDC Giga
de-en	21.9	22.4 (+.5)
cs-en	24.2	25.6 (+1.4)
fr-en	29.1	31.0 (+1.9)
es-en	29.1	30.7 (+1.6)
en-es	31.5	31.8 (+.3)
en-fr	30.3	30.8 (+.5)

Table 3: Using the LDC Gigaword corpora to train larger language models.

LP	Baseline	Big-Tune
de-en	21.4	21.6 (+.2)
fr-en	28.4	28.7 (+.3)
es-en	28.9	29.0 (+.1)
cs-en	23.9	24.1 (+.2)
en-de	15.8	15.9 (+.1)
en-fr	28.7	29.2 (+.5)
en-es	30.9	31.2 (+.2)
en-cs	17.2	17.4 (+.2)

Table 4: Using a larger tuning set (7567 sentences) by combining newstest 2008 to 2010.

4 Better Tuning

4.1 Bigger Tuning Sets

In recent experiments, mainly geared towards using much larger feature sets, we learned that larger tuning sets may give better and more stable results. We tested this hypothesis here as well.

By concatenating the sets from three years (2008-2010), we constructed a tuning set of 7567 sentences per language. Table 4 shows that we gain on average about +0.2 BLEU points.

4.2 Pairwise Ranked Optimization

We recently added an implementation of the pairwise ranked optimization (PRO) tuning method (Hopkins and May, 2011) to Moses as an alternative to Och's (2003) minimum error rate training (MERT). We checked if this method gives us better results. Table 5 shows a mixed picture. PRO gives slightly shorter translations, probably because it optimises sentence rather than corpus BLEU, which has a noticeable effect on the BLEU score. For 2 language pairs we see better results, for 4 worse, and for 1 there is no difference. On other data and language conditions we have observed better and more stable results with PRO.

LP	MERT	PRO	PRO-MERT
de-en	21.7 (1.01)	21.9 (1.00) +.2	21.7 (1.01) ±.0
es-en	29.1 (1.02)	29.1 (1.01) ±.0	29.1 (1.02) ±.0
cs-en	24.2 (1.03)	24.5 (1.00) +.3	24.2 (1.03) ±.0
en-de	16.0 (1.00)	15.7 (0.96) –.3	16.0 (1.00) ±.0
en-fr	29.3 (0.98)	28.9 (0.96) –.4	29.3 (0.98) ±.0
en-es	31.5 (0.98)	31.3 (0.97) –.2	31.4 (0.98) –.1
en-cs	17.4 (0.97)	16.9 (0.92) –.5	17.3 (0.97) –.1

Table 5: Replacing the line search method of MERT with pairwise ranked optimization (PRO).

We tried to use PRO to generate starting points for MERT optimization. Theoretically this will lead to better optimization on the tuning set, since MERT optimization steps on PRO weights will never lead to worse results on the sampled n-best lists. This method (PRO-MERT in the table) applied here, however, did not lead to significantly different results than plain MERT.

5 What did not Work

Not everything we tried worked out. Notably, two promising directions — hierarchical models and semi-supervised learning — did not yield any improvements. It is not clear if we failed or if the methods failed, but we will investigate this further in future work.

5.1 Hierarchical Models

Hierarchical models (Chiang, 2007) have been supported already for a few years by Moses, and they give significantly better performance for Chinese–English over phrase-based models. While we have not yet seen benefits for many other language pairs, the eight language pairs of WMT12 allowed us to compare these two models more extensively, also in view of recent enhancements resulting in better search accuracy.

Since hierarchical models are much larger (roughly 10 times bigger), we trained hierarchical models on downsized training data for most language pairs. For Spanish and French, this excludes UN and GigaFrEn; for Czech some parts of the CzEng corpus were excluded based on their lower language model interpolation weights relative

LP	Phrase	Downsized	Hierarchical
de-en	21.6	same	21.4 (–.2)
fr-en	28.7	27.9	27.6 (–.3)
es-en	29.0	28.9	28.4 (–.5)
cs-en	24.1	22.4	22.0 (–.4)
en-de	15.9	same	15.5 (–.4)
en-fr	29.2	28.8	28.0 (–.8)
en-es	31.2	30.8	30.4 (–.4)
en-cs	17.4	16.2	15.6 (–.6)

Table 6: Hierarchical phrase models vs. baseline phrase-based models.

to their size.

Table 6 shows inferior performance for all language pairs (by about half a BLEU point), although results for German–English are close (–0.2 BLEU).

5.2 Semi-Supervised Learning

Other research groups have reported improvements using semi-supervised learning methods to create synthetic parallel data from monolingual data (Schwenk et al., 2008; Abdul-Rauf and Schwenk, 2009; Bertoldi and Federico, 2009; Lambert et al., 2011). The idea is to translate in-domain monolingual data with a baseline system and filter the result for use as an additional parallel corpus.

Table 7 shows out results when trying to emulate the approach of Lambert et al. (2011). We translate the some of the 2011 monolingual news data (139 million words for French and 100 million words for English) from the target language into the source language with a baseline system trained on Europarl and News Commentary. Adding all the obtained data hurts (except for minimal improvements over a small French-English system). When we filtered out half of the sentences based on translation scores, results were even worse.

Setup	Baseline	+synthetic	+syn-half
fr-en ep+nc	28.0	28.1 (+.1)	28.0 (±.0)
+un	28.7	28.6 (–.1)	28.5 (–.2)
en-fr ep+nc	28.8	28.2 (–.6)	28.1 (–.7)
+un	29.3	28.9 (–.4)	28.9 (–.4)

Table 7: Using semi-supervised methods to add synthetic parallel data to a baseline system trained on Europarl (ep)m News Commentary (nc) and United Nations (un). We added all generated data (synthetic) or filtered out half based on model scores (syn-half).

Acknowledgments

This work was supported by the EuroMatrixPlus project funded by the European Commission (7th Framework Programme).

References

Abdul-Rauf, S. and Schwenk, H. (2009). On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 16–23, Athens, Greece. Association for Computational Linguistics.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Lambert, P., Schwenk, H., Servan, C., and Abdul-Rauf, S. (2011). Investigations on translation model adaptation using monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 284–293, Edinburgh, Scotland. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL)*.

Schwenk, H., Estève, Y., and Rauf, S. A. (2008). The LIUM Arabic/English statistical machine translation system for IWSLT 2008. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 63–68.

Schwenk, H., Lambert, P., Barrault, L., Servan, C., Abdul-Rauf, S., Afli, H., and Shah, K. (2011). Lium's smt machine translation systems for wmt 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 464–469, Edinburgh, Scotland. Association for Computational Linguistics.

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

Using Source-Language Transformations to Address Register Mismatches in SMT

Manny Rayner, Pierrette Bouillon
University of Geneva, FTI/TIM
40 bvd du Pont-d'Arve, CH-1211 Geneva 4, Switzerland
{Emmanuel.Rayner,Pierrette.Bouillon}@unige.ch

Barry Haddow
School of Informatics, The Informatics Forum, 10 Crichton Street
Edinburgh EH8 9AB, University of Edinburgh, Scotland
bhaddow@inf.ed.ac.uk

Abstract

Mismatches between training and test data are a ubiquitous problem for real SMT applications. In this paper, we examine a type of mismatch that commonly arises when translating from French and similar languages: available training data is mostly formal register, but test data may well be informal register. We consider methods for defining surface transformations that map common informal language constructions into their formal language counterparts, or vice versa; we then describe two ways to use these mappings, either to create artificial training data or to pre-process source text at run-time. An initial evaluation performed using crowd-sourced comparisons of alternate translations produced by a French-to-English SMT system suggests that both methods can improve performance, with run-time pre-processing being the more effective of the two.

1 Introduction

The most common problem when doing Statistical Machine Translation in the real world is that there isn't enough data, and the second most common problem is that there isn't enough of the right kind of data; in other words, that there is a mismatch between training and test. In this paper, we will look at an example of a common type of mismatch, which arises within the context of the European Framework ACCEPT project. ACCEPT[1] is concerned with the subject, rapidly growing in importance, of translating the content of online user forums. Given the large variety of possible technical topics and the limited supply of online gurus, it frequently happens that users, searching forum posts online, find that the answer they need is in a language they do not know.

Currently available tools, for example Google Translate, are of course a great deal better than nothing, but still leave much to be desired. When one considers that advice given in an online forum may not be easy to follow even for native language speakers, it is unsurprising that a Google Translated version often fails to be useful. There is consequently strong motivation to develop an infrastructure explicitly designed to produce high-quality translations. ACCEPT intends to achieve this by a combination of three technologies: monolingual pre-editing of the source; domain-tuned SMT; and monolingual post-editing of the target. The manual pre- and post-editing stages are performed by the user communities which typically grow up around online forums. In addition, the SMT stage can optionally be bracketed between automatic pre- and post-editing stages.

In this paper, we will only consider the automatic stages of the translation process in the French-to-English translation pair; we wish to translate French forum data for the benefit of English-speaking users. This rapidly exposes a mismatch between training and test data at the level of register. Forum posts are typically informal in tone. The vast majority of available aligned French/English training data is however formal: a typical example, which we will

[1] Automated Community Content Editing PorTal; http://www.accept.unige.ch.

use in the rest of the paper as our primary resource, is the proceedings of the European parliament, the ubiquitous Europarl corpus (Koehn, 2005). Similar problems would have arisen if we had used other corpora, e.g. the UN corpus[2], Callison-Burch's giga corpus[3] or the Canadian Hansard corpus[4].

French is a language where the gap between formal and informal usage is large. (For purposes of comparison, it is much larger than in English, though perhaps not as large as in Arabic). We will focus on two immediate problems, verb forms and questions. French, like most European languages (English is the major exception) has two second-person pronouns, the formal *vous* and the informal *tu* (accusative form *te*, elided to *t'* before a vowel). Each pronoun has multiple different associated verb inflections. Thus for example the present, future and subjunctive forms of *voir* (to see) are *voyez*, *verrez* and *voyiez* for the formal pronoun, but *vois*, *verras* and *voies* for the informal one. Question-formation is also linked to the distinction between formal and informal register. In the informal register, questions are often formed using the expression *est-ce que*, e.g. *Est-ce que vous avez des pommes?*, "Do you have apples?". In the formal register, the question is usually formed by inversion of subject and verb, so here *Avez-vous des pommes?* If the subject is not a pronoun, this requires introduction of a dummy subject, e.g. *Votre ami a-t-il des pommes?*, "Does your friend have apples?", literally "Your friend does he have apples?"

In some contexts, for example literary translation, it would be important to maintain register differences when translating French to English, perhaps translating *Est-ce que tu veux venir?* as "You wanna come?" but *Voulez-vous venir?* as "Do you want to come?" In the context of forum chat, where the central issue is obtaining useful help, this seems an overrefinement. In what follows, we will assume that we can freely translate informal French constructions as formal English constructions, which will make it much easier to reuse formal-register training data.

[2] http://www.euromatrixplus.net/multi-un/

[3] http://www.statmt.org/wmt10/training-giga-fren.tar

[4] http://www.isi.edu/natural-language/download/hansard/

Table 1 presents figures, contrasting numbers of occurrence of *tu*, *te* and *est-ce que* in the French-English Europarl version 6 corpus (formal register) and ACCEPT forum logs (informal register). As can be seen, *tu*, *te* and *est-ce que* are all common words in ACCEPT, but much less common in Europarl. The limited quantity of training data for informal-register constructions results in problematic translations when they occur in ACCEPT test data; for example, the question-formation phrase *est-ce que* is often translated literally as "is it that", and informal second-person verbs often turn out to be out-of-vocabulary.

	Europarl		ACCEPT	
tu	273	0.015%	5421	6.9%
te	105	0.006%	2154	2.7%
est-ce que	1109	0.061%	212	0.3%
vous	90564	4.9%	4022	5.1%
questions	62031	3.4%	6588	8.4%
#sents	1825077		77819	

Table 1: Numbers of sentences containing occurrences of *tu*, *te*, *est-ce que* and *vous*, as well as numbers of questions, in the French-English Europarl corpus (formal register) and ACCEPT forum logs (informal register).

In the rest of the paper, we will describe experiments in which we have attempted to address these problems by means of source-language rewriting rules which transform informal register constructions to corresponding formal-register ones. The rewriting rules are written in a minimal regular-expression based transduction notation, which only requires access to a good source of lexical information. Transformation rules can be used either at training time or at run-time. At training time, rules can be used to create artificial training data by transforming existing formal-register corpus material into informal-register. Alternately, at run-time, rules working in the opposite direction can be treated as a pre-processing stage, applied before use of the SMT engine, which transforms informal-register phrases into formal-register counterparts. We present the rules themselves and then the results of the experiments.

2 Creating rewriting rules

We begin by considering rules for creating artificial training data. At the end of the section, we briefly consider how to invert these rules to construct rules that can be used at run-time.

2.1 Rules for creating artificial data

Our starting point was the French/English Europarl corpus, which contains 1.8M aligned sentence pairs; we began by writing rules which transformed French sentences not containing the lexical items of interest to us (*tu*, *te* and *est-ce que*) into sentences, equivalent in meaning, which did contain these items. Since the transformed sentence has the same meaning as the original one, it can safely be aligned against the same English sentence, to create more training data.

We obtain the lexical information we need from the MMORPH system (Petitpierre and Russell, 1995). The French MMORPH lexicon contains about 2.1M surface forms, each associated with a feature-value list encoding grammatical information. A typical MMORPH verb entry is the following one for *affirmeriez*, the second person plural conditional form of *affirmer*, "to affirm":

```
"affirmeriez" = "affirmer"
Verb[ mode=conditional
tense=present number=plural
person=2 form=surface
type=1 derivation=ant ]
```

For ease of use, the MMORPH lexicon is converted into Prolog form; to increase efficiency, a little preprocessing is carried out to cache some relationships between surface words, in particular the relationship between corresponding second person singular and plural forms of verbs. Transduction is performed by a simple interpreter, also implemented in Prolog; the interpreter will be released as open source and consists of less than two pages of straightforward code. Rules permit matching of regular expressions, where words can optionally be annotated with Prolog calls to access lexical information.

Figure 1 shows a typical rule, which maps a combination of *vous* and a formal second person plural verb to *tu* and an informal second person singular verb, taking account of possibly accompanying words which may be between the subject and the verb, or immediately after the verb. Reading the rule from top to bottom, we have an occurrence of *vous* replaced by *tu*; an optional negation particle; an optional clitic, which if it is *vous* (reflexive object) is replaced by *te*; an optional second clitic; a formal second person plural verb form, which is replaced by the corresponding informal second person plural verb form using information taken from MMORPH; an optional following verb; and an optional *vous-même*, replaced by *toi-même*.

The only non-obvious aspect of the rule is the element

```
(From:2p_verb_indic(From, To))/To
```

which maps the formal second-person plural verb form `From` to the corresponding informal second-person singular form `To`. For most verb forms, the mapping is unambiguous: thus for example *accueillerez* (second person plural future) maps to *accueilleras* (second person singular future). There is however a systematic ambiguity in the ending `-iez`, which can be either the second person plural present or the second person plural subjunctive. The element specifies that, in the case of an ambiguity, the indicative form should be chosen, as opposed to the subjunctive.

The rule is combined with two similar but more complex rules, which match the common contexts requiring a subjunctive verb and try to map the second-person verb to a subjunctive form if possible. The two subjunctive rules are attempted first, and the indicative one from Figure 1 is only used if they fail. Since subjunctive readings are considerably less frequent than indicative ones, and surface cues for identifying subjunctives are fairly reliable, the combination of the three rules performs well in practice; we will justify this claim in the next section.

The design illustrates both the strengths and the weaknessness of the methodology. On the negative side, the rules integrally depend on *ad hoc* properties of French, in this case the relatively unambiguous second person plural verb inflections and the fact that subjunctive contexts can reliably be predicted using a small set of cue words. Given that these conditions are in fact met, the upside is that we are able to produce a simple set of rules which can be efficiently applied. In general, the methology works if

```
[vous/tu,                                  %vous -> tu
 opt(or(ne, 'n\'')),                       %optional negation
 opt(or(vous/te, (C:clitic(C))/C)),        %optional clitic (vous -> te)
 opt((C1:clitic(C1))/C1),                  %optional 2nd clitic
 (From:2p_verb_indic(From, To))/To,        %2 pl verb (transformed pl to sg)
 or(V:verb(V, _), []),                     %opt verb
 or('vous-même'/'toi-même', [])            %opt vous-même -> toi-même
]
```

Examples:

Je pense que cela a été fait parce que **vous n'en avez** pas parlé. →
Je pense que cela a été fait parce que **tu n'en as** pas parlé.
(I think this has been done because you haven't talked about it)

Vous l'avez dit vous-même, Monsieur le Commissaire.
Tu l'as dit toi-même, Monsieur le Commissaire.
(You have said it yourself, Mr. Commissioner).

Figure 1: Transduction rule for converting second-person plural verbs into corresponding second person singular forms and typical examples of applying the rule, with matched portions in **bold**. Comments are prefaced by percent signs (%). The rule has been slightly simplified for presentational purposes.

it is possible to exploit *ad hoc* properties of this kind to create rules that trigger on reasonably large numbers of sentences that can be transformed into useful variants.

In the experiments described here, we use a total of 12 formal-to-informal rules. In addition to the three described above, we have one rule for creating examples of the accusative informal second-person pronoun *te*, and seven for creating examples of the informal question construction *est-ce que*. The rule for creating examples of *te* is very simple: it matches a sequence consisting of *vous* immediately followed by a verb which is *not* a second-person plural form. This means that the occurrence of *vous* must be an accusative form (*vous* is ambiguous between nominative and accusative, like English *you*), and hence it can here be safely replaced by *te*. The rules for *est-ce que* look for several different versions of sequences, characteristic of questions involving inverted word-order, consisting of a verb followed by a hyphen and a subject pronoun; they rearrange them into corresponding sequences using *est-ce que*, for example mapping *voulez-vous ... ?* ("want-you ... ?") into *est-ce que vous voulez ... ?* ("*est-ce que* you want ... ?")

2.2 Run-time rules

The first group of rules above, which transform *vous* to *tu* with changes to the associated verbs, are easy to reverse, and have almost the same form. The reversed versions, which map *tu* to *vous*, can thus be applied at runtime. Since *te* is unambiguous, the reversed rules can safely be extended so that *te* is also mapped to *vous*, in effect creating a set of rules which reverse and combine the first and second groups.

The rules for *est-ce que* are much less complete, only covering certain specific contexts involving pronouns, and are not straightforward to reverse. We will discuss the issues concerned at the end of § 3.4.

3 Experiments

We conducted two groups of experiments, using both the formal-to-informal and the informal-to-formal sets of rules. In the first group, we apply the formal-to-informal rules to the Europarl corpus create artificial training data, and then retrain the ACCEPT SMT models; in the second group, we use the informal-to-formal rules to pre-process ACCEPT data, and then use the baseline SMT models.

In both cases, intuitive assessment of the results

suggests that use of the rules often has a positive effect, but the difference in BLEU is small. We consequently performed a contrastive evaluation, presenting pairs of differing translations to judges and asking them to mark which of the two candidate translations they preferred. Judges were recruited through the Amazon Mechanical Turk.

In the rest of this section, we first describe the evaluation methodology (§ 3.1), then the creation of the baseline SMT system (§ 3.2), the group of experiments where the rules were used to create artificial training data, (§ 3.3), and finally the group where the rules were used at run-time (§ 3.4).

3.1 Contrastive evaluation using the Amazon Mechanical Turk

To evaluate the difference in performance between two versions of the French-to-English ACCEPT system on a given corpus, we perform the following analysis. We extract all triples ⟨source, translation$_1$, translation$_2$⟩ for which translation$_1$ and translation$_2$ are different. Triples are divided into groups of 20 and posted as HITs on the Amazon Mechanical Turk, offering a payment of $1 per HIT. We limited participation to workers resident in Canada (a country which has both French and English as official languages), requesting only people who were native speakers of English with a good knowledge of written French, and who had moreover already completed at least 50 HITs of which at least 80% had been accepted by the poster. We required three separate judges for each HIT.

Each judge sees the ⟨source, translation$_1$, translation$_2$⟩ displayed with translation$_1$ and translation$_2$ presented in a random order, with all the diverging words highlighted in red. Since the average length of a ACCEPT sentence is about 17 words, but the number of highlighted words in a translation is usually between 1 and 4, this greatly simplifies the judge's task. For each triple, the judge chooses between five possible results: first clearly better, first slightly better, about equal, second slightly better, second clearly better. We aggregated results using majority judgements, scoring the result as "unclear" if there was no majority. We evaluate significance of results by applying the McNemar sign test to the aggregated numbers of "better" and "worse" judgements. To estimate inter-judge agreement, we marked groups of judgements as one of the following:

Unanimous All three judgements were identical.

Agree Either no one preferred the first translation or no one preferred the second translation.

Strong disagree One judge strongly preferred the first translation and one strongly preferred the second.

Weak disagree Remaining cases.

The average judging time for a 20-item HIT was 6 minutes and 15 seconds, corresponding to an hourly rate of $9.63, good payment by AMT standards. The strong inter-annotator agreement figures we present below suggests that judges were pleased with the conditions offered and worked conscientiously. Restricting judges to a bilingual country appears to be important. We tried removing this condition, and obtained faster turnaround time but much poorer-quality results, with weak inter-annotator agreement and many anomalous judgements suggesting that judges lacked fluency in one or the other language or were not taking the job seriously.

3.2 Training Data and SMT Systems

The SMT baseline system was a phrase-based system trained with the standard Moses pipeline (Koehn et al., 2007), using GIZA++ (Och and Ney, 2000) for word alignment and SRILM (Stolcke, 2002) for the estimation of 5-gram Kneser-Ney smoothed (Kneser and Ney, 1995) language models.

For training the translation and lexicalised reordering models we used the releases of europarl and news-commentary provided for the WMT12 shared task (Callison-Burch et al., 2012), together with a dataset from the ACCEPT project consisting mainly of technical product manuals and marketing materials. This last data set covers the same topics as the forums we wish to translate (so it may be considered as "in-domain") but it is almost exclusively in the formal register.

For language modelling we used the target sides of all the parallel data, together with approximately 900,000 words of monolingual English data extracted from web forums of the type that we wish to translate. Separate language models were trained

on each of the data sets, then these were linearly interpolated using SRILM to minimise perplexity on a heldout portion of the forum data.

For tuning and testing, we extracted 1000 sentences randomly from a collection of monolingual French forum data (distinct from the monolingual English forum data), translated these using Google Translate, then post-edited to create references. The post-editing was performed by a native English speaker, who is also fluent in French. This 1000 sentence parallel text was then split into two equal halves (`devtest_a` and `devtest_b`) for minimum error rate tuning (MERT) and testing, respectively.

3.3 Creating artificial training data

In our first group of experiments, we applied all the formal-to-informal rules to the French half of the French/English Europarl corpus, creating about 80K transformed pairs; since all the rules transform sentences into paraphrases of themselves, the English side of the pair can be left unchanged. Table 2 summarises the data produced. The transformation process involves three passes, one for each type of rule, and takes a total of about 15 minutes on a high-end laptop.

Type	#Pairs
tu	37184
te	20926
est-ce que	21814

Table 2: Transformed French/English Europarl data produced by rewriting rules of different types.

We added the new artifically produced data to the existing set and retrained the ACCEPT SMT models using the expanded data set. We created two different retrained models: the first (**TU/TE**) added only the *tu* and *te* corpora, and the second (**EST-CE-QUE**) added only the *est-ce que* data. In order to facilitate comparisons with **BASELINE**, we did not re-run MERT for the **TU/TE** and **EST-CE-QUE** systems; we re-used the weights from the **BASELINE** system.

Initial evaluation using BLEU on a held-out set of 500 French forum sentences gave inconclusive results; BLEU was slightly better for **TU/TE** and slightly worse for **EST-CE-QUE**, but the difference was in neither case statistically significant. Since we were only attempting to improve performance on a set of words that occurred in about 10% of the sentences in the corpus, this was unsurprising. In order to concentrate on the phenomena of interest, we randomly extracted a set of 200 ACCEPT sentences containing *tu* or *te*, and 200 containing *est-ce que*, from the monolingual corpus of French forum sentences, distinct from any of the data sets used so far. We will refer to these two test corpora as `tu_200` and `est_ce_que_200`. We processed each corpus using both **BASELINE** and the appropriate version of the retrained system, and subjected the results to comparative judging using the methodology described in § 3.1. The results are summarised in Tables 3 and 4, where in each case we give the figures for aggregated comparisons and inter-annotator agreement, as defined in § 3.1.

Aggregated judgements	
Judgement	Number
BASELINE better	34
TU/TE better	68
Unclear	8
Same result	90
Significance	$p < 0.002$
Inter-annotator agreement	
Agreement	Number
Unanimous	64
Agree	18
Weak disagree	28
Strong disagree	0

Table 3: Comparison between **BASELINE** and **TU/TE** SMT models on `tu_200` test corpus, judged by three AMT-recruited judges.

As can be seen, the two sets behave quite differently. Table 3 shows a solid improvement for **TU/TE** compared to **BASELINE**, with 68 better against 34 worse; however, Table 4 indicates a slight *decline* for **EST-CE-QUE**, 43 to 53. Examination of the data shows that the **TU/TE** is succeeding primarily because it is able to fill lexical gaps, most obviously second-person verb forms that did not appear in **BASELINE**'s training data. **EST-CE-QUE**, in contrast, is able to fill very few lexical holes. The

Aggregated judgements	
Judgement	Number
BASELINE better	53
EST-CE-QUE better	43
Unclear	10
Same result	94
Significance	(not significant)
Inter-annotator agreement	
Agreement	Number
Unanimous	40
Agree	30
Weak disagree	32
Strong disagree	4

Table 4: Comparison between **BASELINE** and **EST-CE-QUE** SMT models on `est_ce_que_200` test corpus, judged by three AMT-recruited judges.

expression *est-ce que* can usually only be translated well when a substantial amount of context is taken into account. The literal translations "is that" or "is it that" are not completely wrong, and attempts to improve on these often just make things worse; adding more training data with examples of *est-ce que* confuses the system as often as it helps it. Thus, although we find positive examples like[5]:

Source: Est-ce que je peux installer NIS2011?
Trans$_1$: Is that I can install NIS2011?
Trans$_2$: Can I install NIS2011?

we get even more negative ones like:

Source: Est-ce que je dois acheter une licence?
Trans$_1$: Is that I have to purchase a license?
Trans$_2$: Can I must purchase a license?

Although it seems to us that there are still interesting possibilities to explore here, the second person singular/plural transformation holds out more immediate promise of concrete gains, and we consequently decided to focus on it in the second set of experiments.

3.4 Applying rules at run-time

Given that the main effect of the artificially created informal second-person data is to fill lexical holes, and that the relevant transformation rules can readily be reversed, it is natural to investigate the idea of using the reversed rules at run-time (cf. § 2.2). This idea is easy to implement: we apply the reversed rules as part of the automatic pre-processing stage (cf. description of the ACCEPT architecture in § 1), replacing *tu* and *te* with *vous* and changing associated second-person singular verbs to the corresponding second-person plural forms in contexts licensed by the rules. The result is then submitted to the **BASELINE** SMT engine. Table 5 shows the result of performing these operations on the `tu_200` test set, evaluated as before by comparing against the plain result obtained without pre-processing. As can be seen, the margin of improvement (87 to 35) is even greater than the 68–34 given by adding artificial training data (Table 3 above).

As a sanity check, we asked the evaluators to compare the results of applying pre-processing directly against the result of adding artificial training data (Table 6) and also applied pre-processing to the `devtest_b` set (cf. § 3.2), comparing it against the plain result for this set (Table 7). Reassuringly, judges confirm that pre-processing is better than adding artificial data (66–34), and that application of pre-processing rules to the general devtest set produces a small improvement (31–10).

Aggregated judgements	
Judgement	Number
Non-pre-processed better	35
Pre-processed better	87
Unclear	8
Same result	70
Significance	$p < 0.0001$
Inter-annotator agreement	
Agreement	Number
Unanimous	77
Agree	23
Weak disagree	20
Strong disagree	10

Table 5: Comparison between plain and pre-processed versions of `tu_200` test corpus, translated by **BASELINE** SMT model and judged by three AMT-recruited judges.

Finally, it is important to note that the non-trivial contexts which most of the rules possess are essen-

[5] In the following two examples, Trans$_1$ is the translation produced by **BASELINE** and Trans$_2$ that produced by **EST-CE-QUE**.

Aggregated judgements	
Judgement	Number
TU/TE/non-pre-processed better	34
BASELINE/pre-processed better	66
Unclear	12
Same result	88
Significance	$p < 0.002$
Inter-annotator agreement	
Agreement	Number
Unanimous	68
Agree	18
Weak disagree	24
Strong disagree	2

Table 6: Comparison between plain version of `tu_200` test corpus translated by **TU/TE** SMT model and pre-processed versions translated by **BASELINE** SMT model, judged by three AMT-recruited judges.

Aggregated judgements	
Judgement	Number
Non-pre-processed better	10
Pre-processed better	31
Unclear	4
Same result	466
Significance	$p < 0.002$
Inter-annotator agreement	
Agreement	Number
Unanimous	23
Agree	11
Weak disagree	8
Strong disagree	3

Table 7: Comparison between plain and pre-processed versions of `devtest_b` corpus, translated by **BASELINE** SMT model and judged by three AMT-recruited judges.

tial. In order to test this, we constructed a trivial set of informal-to-formal transformation rules, which simply map every second person singular word (*tu*, *te*, second person singular verb forms, etc) to the corresponding second person plural form. The result was very bad, since, without the constraining contexts, the rules seriously overmatch. Table 8 shows a comparison between the rules used in the main experiments (i.e. with context) and the trivial rules without context.

We had originally intended to carry out similar experiments using reversed versions of the rules for *est-ce que*, but initial investigations convinced us that the problems involved are much more challenging in nature. There are two difficulties. First, the formal-to-informal rules we defined for *est-ce que* only work for examples where the subject is a pronoun, which is the minority case; in the `est_ce_que_200` corpus, less than 30% of the examples have a pronominal subject. Second, and even more seriously, the inverted rules would transform *est-ce que* into constructions with an inverted subject, but it is not in fact clear that this transformation improves the quality of SMT translation. Our overall judgement was that a simple approach of the kind we used successfully for tu/vous has almost no chance of succeeding.

4 Conclusions and further directions

Register mismatches are a common problem in SMT, normally arising because training data is formal register and test data is informal register. We have presented an initial study carried out on a French-to-English SMT system, using source-language rewriting rules both to create artificial training data and as a run-time pre-editing step. We created rules for two common constructions typical of informal-register French language: second-person singular verb forms, and question-formation using *est-ce que*.

Perhaps the most interesting aspect of the study is the very different performance we obtained for the two phenomena. For second-person singular verb forms, both creation of artificial training data and run-time pre-processing worked well, with clear improvements on sentences containing these lexical items; run-time pre-processing appears to be the more effective of the two methods. Our guess is that there are many similar cases, both in this language pair and others, which can be handled using similar methods. The prerequisites seem to be the following:

- Existence of equivalent formal-register words that the informal-register words can be replaced by;

Aggregated judgements	
Judgement	Number
Pre-processing with context better	90
Pre-processing w/o context better	27
Unclear	2
Same result	81
Significance	$p < 0.0001$
Inter-annotator agreement	
Agreement	Number
Unanimous	87
Agree	17
Weak disagree	14
Strong disagree	1

Table 8: Comparison between use of pre-processing rules with and without context on `tu_200` test corpus, translated by **BASELINE** SMT model and judged by three AMT-recruited judges.

- Good SMT translation of the formal-register counterparts; and
- Availability of surface patterns that can identify relevant occurrences of the informal-register words.

In particular, it seems reasonable to us to suppose that the methods would port to other source languages which use different verb-forms for formal and informal language.

The successful treatment of second-person singular verb forms, however, contrasts sharply with the completely unsuccessful attempt to use the same methods on *est-ce que*. We used the rules to create about 20K extra aligned pairs of training sentences. Hand-examination of the artificial data showed that it was of good quality, and yet it not only failed to improve the translation of *est-ce que*, but even degraded it slightly. The problem is the non-local and highly context-dependent translation of *est-ce que*; this depends on the following main verb, which may be widely separated from it. Thus, for example[6], in

Est-ce que la destination de sauvegarde est sur un disque externe? →
Is the save destination on an external drive?

the translation of *est-ce que* becomes "Is" because the following verb is *est*, while in

Est-ce que quelque chose vous semble bizarre avec mon réseau? →
Does something seem strange to you about my network?

the translation is "Does...seem", because the following verb is *semble*.

With enough training data, it is possible that an SMT engine would be able to learn these patterns robustly, but our impression is that a great deal of data would be needed. A more promising approach seems to be to write runtime transduction rules, operating both pre- and post-translation, which perform the necessary regularizations of the source and target language word-orders, as for example described in (Nießen and Ney, 2004). We will be investigating this idea in the near future.

[6]The following examples are taken from the `est_ce_que_200` corpus.

Acknowledgements

The work described in this paper was performed as part of the Seventh Framework Programme ACCEPT project, under grant agreement 288769.

References

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia, editors. 2012. *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Montréal, Canada, June.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL Demo Session*.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit 5*.

S. Nießen and H. Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational linguistics*, 30(2):181–204.

F.J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong.

D. Petitpierre and G. Russell. 1995. Mmorph-the multext morphology program. *Multext deliverable report for the task*, 2(1).

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*. ISCA.