Proceedings of the 16th EAMT Conference, 28-30 May 2012, Trento, Italy

Mixture-Modeling with Unsupervised Clusters for Domain Adaptation in Statistical Machine Translation

Rico Sennrich
Institute of Computational Linguistics
University of Zurich
Binzmühlestr. 14
CH-8050 Zürich
sennrich@cl.uzh.ch

Abstract

In Statistical Machine Translation, in-domain and out-of-domain training data are not always clearly delineated. This paper investigates how we can still use mixture-modeling techniques for domain adaptation in such cases. We apply unsupervised clustering methods to split the original training set, and then use mixture-modeling techniques to build a model adapted to a given target domain. We show that this approach improves performance over an unadapted baseline, and several alternative domain adaptation methods.

1 Introduction

As the availability of parallel data for Statistical Machine Translation (SMT) increases, new opportunities and challenges for domain adaptation arise. Some corpora may contain text from a variety of domains, especially if they are built from heterogeneous resources such as crawled web pages. Many domain adaptation techniques do not operate on a single text, but require multiple models which are then mixed.

We investigate domain adaptation in a scenario where we have a known target domain, including development and test data from this domain, but where there is only a single heterogeneous training corpus. While this training corpus does contain in-domain data, we assume that we have no supervised means of extracting it.

Our basic approach is divided into two steps. Firstly, we perform unsupervised clustering on the parallel training data to obtain a given number of clusters. Secondly, we apply domain adaptation algorithms to compute a model from these clusters that is adapted to the development set.

2 Related Work

The general idea in domain adaptation is to obtain models that are specifically optimized for best performance in one domain, with a potentially negative effect on its performance for other domains. The classical domain adaptation scenario consists of a (small) in-domain corpus, a (large) out-of-domain corpus, and in-domain development and test sets. Mixture-modeling approaches such as (Koehn and Schroeder, 2007; Foster and Kuhn, 2007; Sennrich, 2012) fall into this category.

We will here give an overview of adaptation techniques that assume less prior knowledge about the training set and/or target domains.

Yamamoto and Sumita (2008) operate without any predetermined domains, and without assuming that either the training or the test data is homogeneous. They cluster the training text into k clusters, and use unsupervised domain selection to translate each test set sentence by a cluster-specific model.

Finch and Sumita (2008) distinguish between two classes of sentences: questions and declaratives (i.e. non-questions). They split the training corpus automatically according to a simple rule (does the target sentence end with '?'), and for decoding use a linear interpolation of the class-specific and a general model, the interpolation weight depending on the class membership of each sentence.

Banerjee et al. (2010) focus on a scenario in which the domains of the training texts are known, whereas the test sets are a mix of two domains. They use a sentence-level classifier to translate each sentence with a domain-specific SMT system.

$$\overline{K}_m(x,x') = \sum_{n=1}^{m} \sum_{u \in \Sigma^n} \frac{f_x(u)}{\sqrt{\sum_{v \in \Sigma^n} f_x(v)}} \frac{f_{x'}(u)}{\sqrt{\sum_{v \in \Sigma^n} f_{x'}(v)}} \tag{1}$$

$$\overline{K}_m(x,x') = \sum_{n=1}^{m} \sum_{u \in \Sigma^n} \sqrt{\frac{f_x(u)}{\sum_{v \in \Sigma^n} f_x(v)} \frac{f_{x'}(u)}{\sum_{v \in \Sigma^n} f_{x'}(v)}} \tag{2}$$

Eck, Vogel and Waibel (2004) use information retrieval techniques to find the sentences in a parallel corpus that are closest to the translation input, then use the corresponding target sentences to build a language model. Their approach is similar to that of Yamamoto and Sumita (2008) in that both try to adapt models in a fully unsupervised manner. The main difference is that Yamamoto and Sumita (2008) compute the clusters (and the cluster-specific models) offline, and only do cluster prediction online, whereas in (Eck et al., 2004), the whole adaptation process, i.e. selecting a subset of training data, training a model, and translating with the specific model, happens online.

We will focus on a scenario which is slightly different from these prior studies in that we want to build a translation system for a specific target domain, but with in-domain and out-of-domain training data being mixed in a heterogeneous training set. For such a scenario, none of the outlined approaches are a perfect fit. Mixture-modeling techniques presume the existence of multiple models to mix, a condition which is not met in this scenario. The unsupervised methods, on the other hand, do not use sophisticated adaptation techniques, mostly because the target domain is unknown. We will test a hybrid approach that combines unsupervised methods to cluster the training text with known mixture-modeling techniques to obtain a model adapted to the target domain.

3 Clustering

We compare two unsupervised sentence clustering algorithms in order to split the training text into clusters that can later be recombined. Both algorithms are instances of k-means clustering, but with different distance functions. Yamamoto and Sumita (2008) use language models as centroids, trained on all sentences in a cluster, and the language model entropy as the distance between each sentence and cluster. Andrés-Ferrer et al. (2010) use word-sequence-kernels (WSK) (Cancedda et al., 2003) as distance metric between two documents. We initially followed their proposed normalization of the WSK, reproduced in equation 1. $f_x(u)$ is the frequency of the n-gram u in document/sentence x.[1] Unfortunately, the normalization in the proposed equation is flawed and causes a bias towards assigning sentences to the largest cluster. The WSK should be normalized so that the string a is (at least) as similar to itself as to $a\ a$ (if we only consider unigrams). However, $\frac{1}{\sqrt{1}}\frac{1}{\sqrt{1}} < \frac{1}{\sqrt{1}}\frac{2}{\sqrt{2}}$. We use an alternative normalization, shown in equation 2, that has no such numerical bias.

Both algorithms are initialized with randomly generated clusters, and both can be expanded to clustering sentence pairs by taking the sum of the distance on both language sides. In terms of n-gram length, we follow the respective authors' practice, using unigram models for the implementation of (Yamamoto and Sumita, 2008), and $m = 2$ for equation 2. Note that the clustering algorithm has the objective of minimizing LM entropy, whereas the WSK is a similarity function and thus is maximized.

3.1 Exponential Smoothing

One drawback of sentence-level clustering is that cluster assignment is made on the basis of very little information, i.e. the sentence itself. If we assume that the domain of a text does not rapidly change between sentences, it is sensible to consider a larger context for clustering.

We achieve this by using an exponentially decaying score for cluster assignment.[2] In the baseline without exponential decay (equation 3), we assign the sentence pair i to the cluster c that

[1] For the full motivation of the equation, see (Andrés-Ferrer et al., 2010). In short, for all n-grams up to a maximum length m, the kernel sums over the product of their normalized frequency in two given documents.

[2] The most similar use of an exponential decay that we are aware of is by Zhong (2005), who proposes exponential decay to reduce the contribution of history data in a text stream clustering algorithm. However, the exponential decay affects a different component, namely the centroids, and does not serve the same purpose as our proposal.

minimizes the distance (i.e. the LM entropy or the negative WSK score).

$$\hat{c}_i = \arg\min_c d(i, c) \tag{3}$$

In equation 4, the distance of sentence pair i to cluster c is smoothed by the weighted average of the distance of each sentence j to c, with the weight exponentially decaying as the textual distance between i and j increases, and with the decay factor λ determining how fast the weight decays.

$$\hat{c}_i = \arg\min_c \sum_{j=1}^{n} d(j, c) \cdot \lambda^{|i-j|} \tag{4}$$

Note that the equation is two-sided, meaning that both previous and subsequent sentences are considered for the assignment.

Algorithmically, two-sided exponential smoothing only slows down cluster assignment by a constant factor; we do not need to sum over all sentences for each assignment, but can store the weighted distance of all previous sentences in a single variable. Algorithm 1 shows the smoothed assignment step for n sentences and k clusters.

Algorithm 1 Cluster assignment with decay

```
Ensure: $0 \leq decay \leq 1$
let $d(x, y)$ be a distance function for a sentence $x$ and a centroid $y$
let $d\_min[n]$,$d\_curr[n]$,$\hat{c}[n]$ be arrays
set all elements of $d\_min$ to $\infty$
for $c = 0$ to $k$ do
    $cache \leftarrow 0$
    set all elements of $d\_curr$ to 0
    for $i = 0$ to $n$ do
        $cache \leftarrow decay * cache$
        $cache \leftarrow cache + d(i, c)$
        $d\_curr[i] \leftarrow cache$
    end for
    $cache \leftarrow 0$
    for $i = n$ to 0 do
        $cache \leftarrow decay * cache$
        $d\_curr[i] \leftarrow d\_curr[i] + cache$
        if $d\_curr[i] < d\_min[i]$ then
            $d\_min[i] \leftarrow d\_curr[i]$
            $\hat{c}[i] \leftarrow c$
        end if
        $cache \leftarrow cache + d(i, c)$
    end for
end for
```

Note that the decay factor λ determines the extent of smoothing, i.e. how strongly context is taken into account for the assignment of each sentence. A decay factor of 0 corresponds to the unsmoothed sentence-level score (with $0^0 = 1$). With a decay factor of 1, the algorithm returns the same distance for all sentence pairs. We use a decay factor of 0.5 throughout the experiments. This is a relatively fast decay: one third of the score is determined by the sentence itself; two thirds by the sentence and its two neighboring sentences. What decay factor is optimal may depend on the properties of the text, i.e. how quickly documents and/or domains change, so we will not evaluate different decay factors in this paper.

We could extend the algorithm to reset the cache to 0 whenever we cross a known document boundary, and thus implement document-level scoring (with a decay factor of 1), or a hybrid (with a decay factor between 0 and 1). We did not do this since we want to demonstrate that the approach does not require document boundaries in the training text.

Another point to note is that we slightly modify the LM entropy method by normalizing entropy by sentence length, which ensures that longer sentences have no inflated effect on their neighbors' cluster assignment.

4 Model Combination

Having split the training text into clusters, there are various possibilities to exploit them. Yamamoto and Sumita (2008) use each cluster to train a cluster-specific model, which they interpolate with a general model, using a constant interpolation coefficient. Translating a text then consists of predicting the cluster of each sentence, then translating it with this cluster-specific model. If we make the assumption that the test set is relatively homogeneous, with all sentences belonging to the same domain, we can perform a more sophisticated adaptation to this target domain.

One potential shortcoming of the algorithm in (Yamamoto and Sumita, 2008) is that their domain prediction has little information to base its prediction on, and thus may not choose the best cluster. Additionally to predicting the domain for each sentence, we will test a document-level domain prediction, i.e. selecting the cluster with the shortest distance to the whole test set. Even this might be suboptimal if the number of clusters is high. In this case, we can expect relevant data to be distributed over multiple clusters, in which case it might be beneficial to not be restricted to one cluster-specific model.

A second shortcoming is the lack of model optimization. Yamamoto and Sumita (2008) set the interpolation weights between the cluster-specific model and the general one manually after some preliminary experiments, and re-used the model parameters from the general model for all experiments. Specifically, they use linear interpolation with interpolation coefficients of 0.7 and 0.3 for the cluster-specific and the general translation model, respectively, and a log-linear combination for language models, with a slightly lower weight for the domain-specific (0.4) than the general (0.6) model.

Both the inability to consider multiple relevant datasets and the need to manually set model weights can be solved by using automatic mixture-model methods. We will experiment with automatic adaptation methods that use perplexity minimization to produce domain-specific models given a development set from the domain. The first step is again to train cluster-specific translation and language models, which we then recombine into a single adapted model. We use a linear interpolation with the interpolation coefficients set through perplexity minimization for language model and translation model adaptation, which has been demonstrated to be a successful technique in SMT (Foster and Kuhn, 2007). For translation model interpolation, we use the approach described in (Sennrich, 2012), optimizing each translation model feature separately on a parallel development set.

The optimization itself is convex, which means that we can easily apply it to a high number of clusters. The biggest risk is that the weight vector will be overfitted if we optimize it for a high number of small models. Finally, we set new log-linear SMT weights through MERT (Och and Ney, 2003) for each experiment.

5 Experiments

The main questions that we want to answer in our experiments are:

1. How well does unsupervised clustering split a heterogeneous training text according to its domains? How are the results affected by different distance functions and smoothing?

Data set	sentences	words (fr)
Alpine (in-domain)	200k	4 400k
Europarl	1 500k	44 000k
JRC Acquis	1 100k	24 000k
OpenSubtitles v2	2 300k	18 000k
Total train	5 100k	90 400k
Dev (perplexity)	1424	33 000
Dev (MERT)	1000	20 000
Test	991	21 000

Table 1: Parallel data sets for German – French translation task.

2. How much translation quality do we lose or gain from mixture-modeling based on unsupervised clusters, compared to a scenario where we start with multiple domain-specific corpora.

5.1 Data and Methods

We perform the experiments on a German–French data set. The parallel data sets used are listed in table 1. The in-domain corpus is a collection of Alpine Club publications (Volk et al., 2010). As parallel out-of-domain data sets, we use Europarl, a collection of parliamentary proceedings (Koehn, 2005), JRC-Acquis, a collection of legislative texts (Steinberger et al., 2006), and OpenSubtitles v2, a parallel corpus extracted from film subtitles[3] (Tiedemann, 2009).

For language model training, we used the same 90 million word corpus, plus, on the target side, the news corpus from WMT 2011 (appr. 610 million tokens), and appr. 8 million tokens monolingual in-domain data. We used the following language model settings: for clustering, unigram language models. For domain selection, 3-gram language models with Good-Turing smoothing. For translation, 5-gram language models with interpolated Kneser-Ney smoothing. We clustered additional target language data with the method described in (Yamamoto and Sumita, 2008), i.e. one cluster assignment step, starting from the bilingual clusters, and not assigning any sentences which are closest to the general LM.

For the clustering experiments, these data sets are concatenated to simulate a heterogeneous training set. The relative amount of in-domain data in the training sets is 2% (monolingual) and 4% (parallel). Note that this makes success of our method

[3] http://www.opensubtitles.org

more likely than in scenarios where there is no in-domain training data in the training set. We do not claim that any heterogeneous training text is equally suited for domain adaptation.

In (Andrés-Ferrer et al., 2010), clustering quality is measured intrinsically, i.e. by calculating the intra-cluster language model perplexity. In our evaluation, we use an extrinsic evaluation that compares the resulting clusters to the original four parallel datasets. For this evaluation, we assume that clustering is felicitous if it clusters sentences from the same original data set together. We measure this using entropy (equation 5), with N being the total number of sentence pairs and $orig(i)$ being the corpus to which sentence i originally belonged. $p_c(orig(i))$ is the probability that a sentence in cluster c is originally from corpus $orig(i)$, estimated through relative frequency.

$$H(X) = -\sum_{c=0}^{k} \sum_{i \in c} \frac{1}{N} \log_2 p_c(orig(i)) \qquad (5)$$

If a cluster only contains sentences from one corpus, its entropy is 0. The baseline is a uniform distribution, which corresponds to an entropy of 1.698 (with the data sets from table 1).

The second evaluation is a translation task. In terms of tools and techniques used, we mostly adhere to the work flow described for the WMT 2011 baseline system[4]. The main tools are Moses (Koehn et al., 2007), SRILM (Stolcke, 2002), and GIZA++ (Och and Ney, 2003), with settings as described in the WMT 2011 guide. One exception is that we additionally filter the phrase table according to statistical significance tests, as described by (Johnson et al., 2007). We use two different development sets, one for domain adaptation (through perplexity optimization) and one for MERT, in order to rule out that MERT gives too much weight to the language and translation model which are optimized on the same dataset.

We measure translation performance through BLEU (Papineni et al., 2002) and METEOR 1.3 (Denkowski and Lavie, 2011). All results are lowercased and tokenized, measured with five independent runs of MERT (Och and Ney, 2003). We perform significance testing with MultEval (Clark et al., 2011), which uses approximate randomization to account for optimizer instability. Note that there are other causes of instability unaccounted for, e.g. the randomness of clustering. Word alignment has been kept constant across all experiments.

distance	k	entropy		itr.
		mean	stdev	(avg)
no smoothing				
WSK	10	0.727	0.022	21.4
LM	10	0.439	0.034	20.2
LM	100	0.344	0.008	38.8
exponential smoothing				
WSK	10	0.263	0.048	13.8
LM	10	0.112	0.016	10.4
LM	100	0.064	0.013	9.0

Table 2: Entropy comparison between clustering with different distance functions (with or without smoothing), and different numbers of clusters (k). Mean, standard deviation, and average number of iterations out of 5 runs are reported. WSK: word sequence kernels; LM: language model entropy

5.2 Results

In all experiments, we perform k-means clustering with $k = 10$ and $k = 100$. A higher number of clusters typically increases the homogeneity of the resulting clusters, and may boost performance by allowing us to give high weights to very specific subdomains of the training set. On the downside, clusters will be smaller on average, which exacerbates data sparseness problems. In the trivial case, having one sentence per cluster results in an entropy of 0, but this granularity would be unsuitable for the domain adaptation methods that we evaluate because of data sparseness.

Table 2 shows entropy of both sentence-level clustering and exponential smoothing with word sequence kernels and LM entropy as distance functions. All methods achieve a strong reduction of entropy over the uniform baseline (1.698), but LM entropy as a distance measure outperforms word sequence kernels, with a mean entropy of 0.439 compared to 0.727 for 10 clusters. In all experiments, exponential smoothing reduces the entropy of the resulting clusters even further. With LM entropy as distance function, it is reduced from 0.439 to 0.112 for $k = 10$, and from 0.344 to 0.064 with $k = 100$. A second advantage of smoothing is that the algorithm converges faster, and reduces the number of iterations by a factor of 2–4. Thus, smoothing seems a good choice because

[4] http://www.statmt.org/wmt11/baseline.html

system	BLEU	METEOR
general	18.5	37.3
adapted TM	18.8	37.8
adapted LM	18.8	37.8
adapted TM & LM	18.6	37.9

Table 3: Baseline SMT results DE–FR. Concatenation of all data and using domain adaptation with original four datasets.

the smoothed algorithm is both faster and better at clustering sentences from the same original dataset into the same cluster. Whether this leads to better SMT performance is tested in the evaluation of translation performance.

We can compare translation performance to four baselines, shown in table 3. The general system (without domain adaptation) performs worst, with a BLEU score of 18.5 and a METEOR score of 37.3. Both TM and LM adaptation significantly increase scores by 0.3 BLEU and 0.5 METEOR points. The system that combines TM and LM adaptation is not significantly different from the systems with only one model adapted in terms of BLEU, but performs best in terms of METEOR (0.6 points better than the general model).

For the experimental systems, we limit ourselves to LM entropy as distance function, and vary a number of parameters. k, the number of clusters, is 10 in table 4, and 100 in table 5. For both k, we test clustering without smoothing (sentence-level clustering) and with exponential smoothing and a decay factor of 0.5. For each variation of these parameters, we pick a single clustering run at random. For model combination, we contrast the approach by Yamamoto and Sumita (2008) (i.e. domain prediction with a fixed interpolation), and the mixture models described in section 4, i.e. perplexity-minimization to find the optimal weights for the linear interpolation of the language and translation model (Sennrich, 2012).

In sections 3.1 and 4, we have identified possible shortcomings of the original approach by (Yamamoto and Sumita, 2008), and will now reiterate and discuss them.

Firstly, we have hypothesized that unsmoothed sentence-level clustering may fail to cluster in-domain data together, and have proposed exponential smoothing. The entropy results in table 2 support this hypothesis; if we look at translation results with document-level domain prediction, the performance differences are small. A look at the clusters that are selected in domain prediction shows that smoothing improved homogeneity (180 000 in-domain / 20 000 out-of-domain sentence pairs) over an unsmoothed sentence-level clustering (146 000 in-domain / 90 000 out-of-domain), but both approaches cluster the majority of the 200 000 in-domain sentence pairs together and outperform the unadapted baseline.

Secondly, we suspected that domain prediction on a sentence-level would suffer from similar data-sparseness problems, and not pick the optimal cluster for translation. With 10 clusters, there is little difference between sentence-level and document-level domain prediction, both in terms of performance and the cluster that is predicted in domain prediction. With (smoothed or unsmoothed) sentence-level prediction, 80-90% of test set sentences are predicted to belong to the same cluster. With 100 clusters, the opposite of our hypothesis is true. Document-level domain prediction performs worse than (smoothed or unsmoothed) sentence-level domain prediction, and no better than the unadapted baseline. For the interpretation of this result, we must also consider the mixture-modeling results.

Adapting models through perplexity optimization performs better than or equally well as the methods with domain prediction and a fixed interpolation between the domain-specific and the general model. This is true for both domain prediction methods, and both smoothed and unsmoothed clustering. The best result is obtained with $k = 10$ and smoothed clustering, with a BLEU score of 19.2 and a METEOR score of 38.3, which is 0.7 BLEU points and 1 METEOR points above the unadapted baseline. The system also beats the adapted baseline, which uses the same model combination algorithm on the original four datasets, by 0.6 BLEU points and 0.4 METEOR points, and the approach by (Yamamoto and Sumita, 2008) (sentence-level clustering and domain prediction) by 0.3 BLEU points and 0.4 METEOR points.

With 100 clusters, perplexity minimization yields no further performance gains, but remains significantly better than the systems with domain prediction and the baseline systems. As to the reason why document-level domain prediction performs poorly with 100 clusters, the main problem is that relevant data is spread out over multiple clusters, and that only a small amount of relevant

clustering	domain prediction	model combination	adapted TM		adapted TM & LM	
			BLEU	METEOR	BLEU	METEOR
sentence-level	sentence-level	fixed weights	18.7	37.6	18.9	37.9
	document-level	fixed weights	18.8	37.7	18.9	37.9
	-	perplexity	18.8	38.0	18.9	38.2
smoothed	smoothed	fixed weights	18.9	37.8	19.0	38.0
	document-level	fixed weights	18.9	37.8	19.0	38.1
	-	perplexity	19.1	38.3	19.2	38.3

Table 4: SMT results DE–FR based on clustered training data ($k = 10$).

clustering	domain prediction	model combination	adapted TM		adapted TM & LM	
			BLEU	METEOR	BLEU	METEOR
sentence-level	sentence-level	fixed weights	18.8	37.7	18.6	37.6
	document-level	fixed weights	18.5	37.5	18.5	37.5
	-	perplexity	19.0	38.0	19.0	38.3
smoothed	smoothed	fixed weights	18.6	37.5	18.5	37.5
	document-level	fixed weights	18.6	37.5	18.4	37.4
	-	perplexity	19.1	38.1	19.1	38.2

Table 5: SMT results DE–FR based on clustered training data ($k = 100$).

data can be considered with document-level domain prediction. Sentence-level domain prediction avoids this problem by choosing different cluster-specific models to translate different sentences, the perplexity mixture-models by being able to give high weights to multiple cluster-specific models.

6 Conclusion

We demonstrate that it is possible to apply mixture-modeling techniques to models that are obtained through unsupervised clustering of a heterogeneous training text. We obtained a modest performance boost from applying mixture-modeling on the clusters rather than the original parallel corpora. The main advantage of the clustering step, however, is that it reduces the requirements for mixture-modeling, eliminating the need for a homogeneous, in-domain training corpus, and only requiring a development set from the target domain. It is thus more general and could be applied to monolithic, heterogeneous data collections.

Compared to the fully unsupervised method by (Yamamoto and Sumita, 2008), we observed small performance improvements of up to 0.3 BLEU points. In a closed-domain setting, the approach also has the advantage of moving the domain adaptation cost into the offline phase, and not requiring a domain prediction phase and multiple models during decoding. To support multiple target domains, the approach could be combined with that of (Banerjee et al., 2010), who discuss the problem of translating texts that contain sentences from multiple (known) domains.

We also propose exponential smoothing during cluster assignment to better capture slow-changing textual properties such as their domain membership, and to combat data sparseness issues when having to do an assignment decision based on short sentences. While the effects on our translation experiments were small, the increased homogeneity of the resulting clusters and the faster speed of convergence indicate that smoothing is a beneficial enhancement to sentence-level k-means clustering.

Acknowledgments

This research was funded by the Swiss National Science Foundation, grant 105215_126999.

References

Andrés-Ferrer, Jesús, Germán Sanchis-Trilles, and Francisco Casacuberta. 2010. Similarity word-sequence kernels for sentence clustering. In *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, pages 610–619, Berlin, Heidelberg. Springer-Verlag.

Banerjee, Pratyush, Jinhua Du, Baoli Li, Sudip Kumar Naskar, Andy Way, and Josef Van Genabith. 2010.

Combining multi-domain statistical machine translation models using automatic classifiers. In *9th Conference of the Association for Machine Translation in the Americas (AMTA 2010)*.

Cancedda, Nicola, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. 2003. Word sequence kernels. *J. Mach. Learn. Res.*, 3:1059–1082, March.

Clark, Jonathan H., Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

Denkowski, Michael and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

Eck, Matthias, Stephan Vogel, and Alex Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In *4th International Conference on Languages Resources and Evaluation (LREC 2004)*.

Finch, Andrew and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 208–215, Stroudsburg, PA, USA. Association for Computational Linguistics.

Foster, George and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.

Johnson, Howard, Joel Martin, George Foster, and Roland Kuhn. 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, Prague, Czech Republic, June. Association for Computational Linguistics.

Koehn, Philipp and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 224–227, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, 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, June. Association for Computational Linguistics.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Sennrich, Rico. 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.

Steinberger, Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*.

Stolcke, A. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, CO, USA.

Tiedemann, Jörg. 2009. News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Volk, Martin, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer, and Beni Ruef. 2010. Challenges in building a multilingual alpine heritage corpus. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Yamamoto, Hirofumi and Eiichiro Sumita. 2008. Bilingual cluster based models for statistical machine translation. *IEICE - Trans. Inf. Syst.*, E91-D:588–597, March.

Zhong, S. 2005. Efficient streaming text clustering. *Neural Networks*, 18(5-6):790–798, July.