Domain Adaptation via Pseudo In-Domain Data Selection

Amittai Axelrod
University of Washington
Seattle, WA 98105
amittai@uw.edu

Xiaodong He
Microsoft Research
Redmond, WA 98052
xiaohe@microsoft.com

Jianfeng Gao
Microsoft Research
Redmond, WA 98052
jfgao@microsoft.com

Abstract

We explore efficient domain adaptation for the task of statistical machine translation based on extracting sentences from a large general-domain parallel corpus that are most relevant to the target domain. These sentences may be selected with simple cross-entropy based methods, of which we present three. As these sentences are not themselves identical to the in-domain data, we call them *pseudo in-domain* subcorpora. These subcorpora – 1% the size of the original – can then used to train small domain-adapted Statistical Machine Translation (SMT) systems which outperform systems trained on the entire corpus. Performance is further improved when we use these domain-adapted models in combination with a true in-domain model. The results show that more training data is not always better, and that best results are attained via proper domain-relevant data selection, as well as combining in- and general-domain systems during decoding.

1 Introduction

Statistical Machine Translation (SMT) system performance is dependent on the quantity and quality of available training data. The conventional wisdom is that more data is better; the larger the training corpus, the more accurate the model can be.

The trouble is that – except for the few all-purpose SMT systems – there is never enough training data that is directly relevant to the translation task at hand. Even if there is no formal genre for the text to be translated, any coherent translation task will have its own argot, vocabulary or stylistic preferences, such that the corpus characteristics will necessarily deviate from any all-encompassing model of language. For this reason, one would prefer to use more in-domain data for training. This would empirically provide more accurate lexical probabilities, and thus better target the task at hand. However, parallel in-domain data is usually hard to find[1], and so performance is assumed to be limited by the quantity of domain-specific training data used to build the model. Additional parallel data can be readily acquired, but at the cost of specificity: either the data is entirely unrelated to the task at hand, or the data is from a broad enough pool of topics and styles, such as the web, that any use this corpus may provide is due to its size, and not its relevance.

The task of domain adaptation is to translate a text in a particular (target) domain for which only a small amount of training data is available, using an MT system trained on a larger set of data that is not restricted to the target domain. We call this larger set of data a *general-domain* corpus, in lieu of the standard yet slightly misleading *out-of-domain* corpus, to allow a large uncurated corpus to include some text that may be relevant to the target domain.

Many existing domain adaptation methods fall into two broad categories. Adaptation can be done at the corpus level, by selecting, joining, or weighting the datasets upon which the models (and by extension, systems) are trained. It can be also achieved at the model level by combining multiple translation or language models together, often in a weighted manner. We explore both categories in this work.

[1] Unless one dreams of translating parliamentary speeches.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 355–362,
Edinburgh, Scotland, UK, July 27–31, 2011.

First, we present three methods for ranking the sentences in a general-domain corpus with respect to an in-domain corpus. A cutoff can then be applied to produce a very small–yet useful– subcorpus, which in turn can be used to train a domain-adapted MT system. The first two data selection methods are applications of language-modeling techniques to MT (one for the first time). The third method is novel and explicitly takes into account the bilingual nature of the MT training corpus. We show that it is possible to use our data selection methods to subselect less than 1% (or discard 99%) of a large general training corpus and still increase translation performance by nearly 2 BLEU points.

We then explore how best to use these selected subcorpora. We test their combination with the in-domain set, followed by examining the subcorpora to see whether they are actually in-domain, out-of-domain, or something in between. Based on this, we compare translation model combination methods.

Finally, we show that these tiny translation models for model combination can improve system performance even further over the current standard way of producing a domain-adapted MT system. The resulting process is lightweight, simple, and effective.

2 Related Work

2.1 Training Data Selection

An underlying assumption in domain adaptation is that a general-domain corpus, if sufficiently broad, likely includes some sentences that could fall within the target domain and thus should be used for training. Equally, the general-domain corpus likely includes sentences that are so unlike the domain of the task that using them to train the model is probably more harmful than beneficial. One mechanism for domain adaptation is thus to select only a portion of the general-domain corpus, and use only that subset to train a complete system.

The simplest instance of this problem can be found in the realm of language modeling, using perplexity-based selection methods. The sentences in the general-domain corpus are scored by their perplexity score according to an in-domain language model, and then sorted, with only the lowest ones being retained. This has been done for language modeling, including by Gao et al (2002), and more recently by Moore and Lewis (2010). The ranking of the sentences in a general-domain corpus according to in-domain perplexity has also been applied to machine translation by both Yasuda et al (2008), and Foster et al (2010). We test this approach, with the difference that we simply use the source side perplexity rather than computing the geometric mean of the perplexities over both sides of the corpus. We also reduce the size of the training corpus far more aggressively than Yasuda et al's 50%. Foster et al (2010) do not mention what percentage of the corpus they select for their *IR-baseline*, but they concatenate the data to their in-domain corpus and report a decrease in performance. We both keep the models separate and reduce their size.

A more general method is that of (Matsoukas et al., 2009), who assign a (possibly-zero) weight to each sentence in the large corpus and modify the empirical phrase counts accordingly. Foster et al (2010) further perform this on extracted phrase pairs, not just sentences. While this soft decision is more flexible than the binary decision that comes from including or discarding a sentence from the subcorpus, it does not reduce the size of the model and comes at the cost of computational complexity as well as the possibility of overfitting. Additionally, the most effective features of (Matsoukas et al., 2009) were found to be meta-information about the source documents, which may not be available.

Another perplexity-based approach is that taken by Moore and Lewis (2010), where they use the cross-entropy difference as a ranking function rather than just cross-entropy. We apply this criterion for the first time to the task of selecting training data for machine translation systems. We furthermore extend this idea for MT-specific purposes.

2.2 Translation Model Combination

In addition to improving the performance of a single general model with respect to a target domain, there is significant interest in using two translation models, one trained on a larger general-domain corpus and the other on a smaller in-domain corpus, to translate in-domain text. After all, if one has access to an in-domain corpus with which to select data from a general-domain corpus, then one might as well use the in-domain data, too. The expectation is that the larger general-domain model should dominate in regions where the smaller in-domain model lacks coverage due to sparse (or non-existent) ngram counts. In practice, most practical systems also perform target-side language model adaptation (Eck et al., 2004); we eschew this in order to isolate the effects of translation model adaptation alone.

Directly concatenating the phrase tables into one larger one isn't strongly motivated; identical phrase pairs within the resulting table can lead to unpredictable behavior during decoding. Nakov (2008) handled identical phrase pairs by prioritizing the source tables, however in our experience identical entries in phrase tables are not very common when comparing across domains. Foster and Kuhn (2007) interpolated the in- and general-domain phrase tables together, assigning either linear or log-linear weights to the entries in the tables before combining overlapping entries; this is now standard practice.

Lastly, Koehn and Schroeder (2007) reported improvements from using multiple decoding paths (Birch et al., 2007) to pass both tables to the Moses SMT decoder (Koehn et al., 2003), instead of directly combining the phrase tables to perform domain adaptation. In this work, we directly compare the approaches of (Foster and Kuhn, 2007) and (Koehn and Schroeder, 2007) on the systems generated from the methods mentioned in Section 2.1.

3 Experimental Framework

3.1 Corpora

We conducted our experiments on the International Workshop on Spoken Language Translation (IWSLT) Chinese-to-English DIALOG task [2], consisting of transcriptions of conversational speech in a travel setting. Two corpora are needed for the adaptation task. Our in-domain data consisted of the IWSLT corpus of approximately 30,000 sentences in Chinese and English. Our general-domain corpus was 12 million parallel sentences comprising a variety of publicly available datasets, web data, and private translation texts. Both the in- and general-domain corpora were identically segmented (in Chinese) and tokenized (in English), but otherwise unprocessed. We evaluated our work on the 2008 IWSLT spontaneous speech Challenge Task[3] test set, consisting of 504 Chinese sentences with 7 English reference translations each. This is the most recent IWSLT test set for which the reference translations are available.

3.2 System Description

In order to highlight the data selection work, we used an out-of-the-box Moses framework using GIZA++ (Och and Ney, 2003) and MERT (Och, 2003) to train and tune the machine translation systems. The only exception was the phrase table for the large out-of-domain system trained on 12m sentence pairs, which we trained on a cluster using a word-dependent HMM-based alignment (He, 2007). We used the Moses decoder to produce all the system outputs, and scored them with the NIST `mt-eval31a` [4] tool used in the IWSLT evalutation.

3.3 Language Models

Our work depends on the use of language models to rank sentences in the training corpus, in addition to their normal use during machine translation tuning and decoding. We used the SRI Language Modeling Toolkit (Stolcke, 2002) was used for LM training in all cases: corpus selection, MT tuning, and decoding. We constructed 4gram language models with interpolated modified Kneser-Ney discounting (Chen and Goodman, 1998), and set the Good-Turing threshold to 1 for trigrams.

3.4 Baseline System

The in-domain baseline consisted of a translation system trained using Moses, as described above, on the IWSLT corpus. The resulting model had a phrase table with 515k entries. The general-domain baseline was substantially larger, having been trained on 12 million sentence pairs, and had a phrase table containing 1.5 billion entries. The BLEU scores of the baseline single-corpus systems are in Table 1.

Corpus	Phrases	Dev	Test
IWSLT	515k	45.43	37.17
General	1,478m	42.62	40.51

Table 1: Baseline translation results for in-domain and general-domain systems.

[2] `http://iwslt2010.fbk.eu/node/33`

[3] Correct-Recognition Result (CRR) condition

[4] `http://www.itl.nist.gov/iad/mig/tools/`

4 Training Data Selection Methods

We present three techniques for ranking and selecting subsets of a general-domain corpus, with an eye towards improving overall translation performance.

4.1 Data Selection using Cross-Entropy

As mentioned in Section 2.1, one established method is to rank the sentences in the general-domain corpus by their perplexity score according to a language model trained on the small in-domain corpus. This reduces the perplexity of the general-domain corpus, with the expectation that only sentences similar to the in-domain corpus will remain. We apply the method to machine translation, even though perplexity reduction has been shown to not correlate with translation performance (Axelrod, 2006). For this work we follow the procedure of Moore and Lewis (2010), which applies the cosmetic change of using the cross-entropy rather than perplexity.

The perplexity of some string s with empirical n-gram distribution p given a language model q is:

$$2^{-\sum_x p(x) \log q(x)} = 2^{H(p,q)} \tag{1}$$

where $H(p,q)$ is the *cross-entropy* between p and q. We simplify this notation to just $H_I(s)$, meaning the cross-entropy of string s according to a language model LM_I which has distribution q. Selecting the sentences with the lowest perplexity is therefore equivalent to choosing the sentences with the lowest cross-entropy according to the in-domain language model. For this experiment, we used a language model trained (using the parameters in Section 3.3) on the Chinese side of the IWSLT corpus.

4.2 Data Selection using Cross-Entropy Difference

Moore and Lewis (2010) also start with a language model LM_I over the in-domain corpus, but then further construct a language model LM_O of similar size over the general-domain corpus. They then rank the general-domain corpus sentences using:

$$H_I(s) - H_O(s) \tag{2}$$

and again taking the lowest-scoring sentences. This criterion biases towards sentences that are both *like* the in-domain corpus and *unlike* the average of the general-domain corpus. For this experiment we re-used the in-domain LM from the previous method, and trained a second LM on a random subset of 35k sentences from the Chinese side of the general corpus, except using the same vocabulary as the in-domain LM.

4.3 Data Selection using Bilingual Cross-Entropy Difference

In addition to using these two monolingual criteria for MT data selection, we propose a new method that takes in to account the bilingual nature of the problem. To this end, we sum cross-entropy difference over each side of the corpus, both source and target:

$$[H_{I-src}(s) - H_{O-src}(s)] + [H_{I-tgt}(s) - H_{O-tgt}(s)] \tag{3}$$

Again, lower scores are presumed to be better. This approach reuses the source-side language models from Section 4.2, but requires similarly-trained ones over the English side. Again, the vocabulary of the language model trained on a subset of the general-domain corpus was restricted to only cover those tokens found in the in-domain corpus, following Moore and Lewis (2010).

5 Results of Training Data Selection

The baseline results show that a translation system trained on the general-domain corpus outperforms a system trained on the in-domain corpus by over 3 BLEU points. However, this can be improved further. We used the three methods from Section 4 to identify the best-scoring sentences in the general-domain corpus.

We consider three methods for extracting domain-targeted parallel data from a general corpus: source-side cross-entropy (Cross-Ent), source-side cross-entropy difference (Moore-Lewis) from (Moore and Lewis, 2010), and bilingual cross-entropy difference (bML), which is novel.

Regardless of method, the overall procedure is the same. Using the scoring method, We rank the individual sentences of the general-domain corpus, select only the top N. We used the top $N = \{35k, 70k, 150k\}$ sentence pairs out of the 12 million in the general corpus [5]. The net effect is that of domain adaptation via threshhold filtering. New MT systems were then trained solely on these small subcorpora, and compared against the baseline model trained on the entire 12m-sentence general-domain corpus. Table 2 contains BLEU scores of the systems trained on subsets of the general corpus.

Method	Sentences	Dev	Test
General	12m	42.62	40.51
Cross-Entropy	35k	39.77	40.66
Cross-Entropy	70k	40.61	**42.19**
Cross-Entropy	150k	42.73	**41.65**
Moore-Lewis	35k	36.86	40.08
Moore-Lewis	70k	40.33	39.07
Moore-Lewis	150k	41.40	40.17
bilingual M-L	35k	39.59	**42.31**
bilingual M-L	70k	40.84	**42.29**
bilingual M-L	150k	42.64	**42.22**

Table 2: Translation results using only a subset of the general-domain corpus.

All three methods presented for selecting a subset of the general-domain corpus (Cross-Entropy, Moore-Lewis, bilingual Moore-Lewis) could be used to train a state-of-the-art machine translation system. The simplest method, using only the source-side cross-entropy, was able to outperform the general-domain model when selecting 150k out of 12 million sentences. The other monolingual method, source-side cross-entropy difference, was able to perform nearly as well as the general-domain model with only 35k sentences. The bilingual Moore-Lewis method proposed in this paper works best, consistently boosting performance by 1.8 BLEU while using less than 1% of the available training data.

5.1 Pseudo In-Domain Data

The results in Table 2 show that all three methods (Cross-Entropy, Moore-Lewis, bilingual Moore-Lewis) can extract subsets of the general-domain corpus that are useful for the purposes of statistical machine translation. It is tempting to describe these as methods for finding in-domain data hidden in a general-domain corpus. Alas, this does not seem to be the case.

We trained a baseline language model on the in-domain data and used it to compute the perplexity of the same (in-domain) held-out dev set used to tune the translation models. We extracted the top N sentences using each ranking method, varying N from 10k to 200k, and then trained language models on these subcorpora. These were then used to also compute the perplexity of the same held-out dev set, shown below in Figure 1.

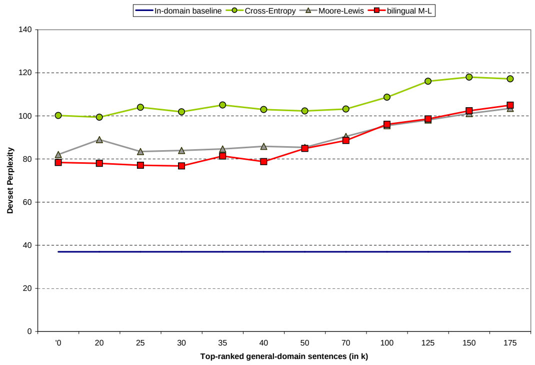

Figure 1: Corpus Selection Results

The perplexity of the dev set according to LMs trained on the top-ranked sentences varied from 77 to 120, depending on the size of the subset and the method used. The Cross-Entropy method was consistently worse than the others, with a best perplexity of 99.4 on 20k sentences, and bilingual Moore-Lewis was consistently the best, with a lowest perplexity of 76.8. And yet, none of these scores are anywhere near the perplexity of 36.96 according to the LM trained only on in-domain data.

From this it can be deduced that the selection methods are not finding data that is strictly in-domain. Rather they are extracting **pseudo in-domain** data which is relevant, but with a differing distribution than the original in-domain corpus.

As further evidence, consider the results of concatenating the in-domain corpus with the best extracted subcorpora (using the bilingual Moore-Lewis method), shown in Table 3. The change in

[5] Roughly 1x, 2x, and 4x the size of the in-domain corpus.

both the dev and test scores appears to reflect dissimilarity in the underlying data. Were the two datasets more alike, one would expect the models to reinforce each other rather than cancel out.

Method	Sentences	Dev	Test
IWSLT	30k	45.43	37.17
bilingual M-L	35k	39.59	**42.31**
bilingual M-L	70k	40.84	**42.29**
bilingual M-L	150k	42.64	**42.22**
IWSLT + bi M-L	35k	47.71	41.78
IWSLT + bi M-L	70k	47.80	**42.30**
IWSLT + bi M-L	150k	48.44	42.01

Table 3: Translation results concatenating the in-domain and pseudo in-domain data to train a single model.

6 Translation Model Combination

Because the pseudo in-domain data should be kept separate from the in-domain data, one must train multiple translation models in order to advantageously use the general-domain corpus. We now examine how best to combine these models.

6.1 Linear Interpolation

A common approach to managing multiple translation models is to interpolate them, as in (Foster and Kuhn, 2007) and (Lü et al., 2007). We tested the linear interpolation of the in- and general-domain translation models as follows: Given one model which assigns the probability $P_1(t|s)$ to the translation of source string s into target string t, and a second model which assigns the probability $P_2(t|s)$ to the same event, then the interpolated translation probability is:

$$P(t|s) = \lambda P_1(t|s) + (1 - \lambda)P_2(t|s) \qquad (4)$$

Here λ is a tunable weight between 0 and 1, which we tested in increments of 0.1. Linear interpolation of phrase tables was shown to improve performance over the individual models, but this still may not be the most effective use of the translation models.

6.2 Multiple Models

We next tested the approach in (Koehn and Schroeder, 2007), passing the two phrase tables directly to the decoder and tuning a system using both phrase tables in parallel. Each phrase table receives a separate set of weights during tuning, thus this combined translation model has more parameters than a normal single-table system.

Unlike (Nakov, 2008), we explicitly did not attempt to resolve any overlap between the phrase tables, as there is no need to do so with the multiple decoding paths. Any phrase pairs appearing in both models will be treated separately by the decoder. However, the exact overlap between the phrase tables was tiny, minimizing this effect.

6.3 Translation Model Combination Results

Table 4 shows baseline results for the in-domain translation system and the general-domain system, evaluated on the in-domain data. The table also shows that linearly interpolating the translation models improved the overall BLEU score, as expected. However, using multiple decoding paths, and no explicit model merging at all, produced even better results, by 2 BLEU points over the best individual model and 1.3 BLEU over the best interpolated model, which used $\lambda = 0.9$.

System	Dev	Test
IWSLT	45.43	37.17
General	42.62	40.51
Interpolate IWSLT, General	48.46	41.28
Use both IWSLT, General	49.13	**42.50**

Table 4: Translation model combination results

We conclude that it can be more effective to not attempt translation model adaptation directly, and instead let the decoder do the work.

7 Combining Multi-Model and Data Selection Approaches

We presented in Section 5 several methods to improve the performance of a single general-domain translation system by restricting its training corpus on an information-theoretic basis to a very small number of sentences. However, Section 6.3 shows that using two translation models over all the available data (one in-domain, one general-domain) outperforms any single individual translation model so far, albeit only slightly.

Method	Dev	Test
IWSLT	45.43	37.17
General	42.62	40.51
both IWSLT, General	49.13	42.50
IWSLT, Moore-Lewis 35k	48.51	40.38
IWSLT, Moore-Lewis 70k	49.65	40.45
IWSLT, Moore-Lewis 150k	49.50	41.40
IWSLT, bi M-L 35k	48.85	39.82
IWSLT, bi M-L 70k	49.10	**43.00**
IWSLT, bi M-L 150k	49.80	**43.23**

Table 5: Translation results from using in-domain and pseudo in-domain translation models together.

It is well and good to use the in-domain data to select pseudo in-domain data from the general-domain corpus, but given that this requires access to an in-domain corpus, one might as well use it. As such, we used the in-domain translation model alongside translation models trained on the subcorpora selected using the Moore-Lewis and bilingual Moore-Lewis methods in Section 4. The results are in Table 5.

A translation system trained on a pseudo in-domain subset of the general corpus, selected with the bilingual Moore-Lewis method, can be further improved by combining with an in-domain model. Furthermore, this system combination works better than the conventional multi-model approach by up to 0.7 BLEU on both the dev and test sets.

Thus a domain-adapted system comprising two phrase tables trained on a total of 180k sentences outperformed the standard multi-model system which was trained on 12 million sentences. This tiny combined system was also 3+ points better than the general-domain system by itself, and 6+ points better than the in-domain system alone.

8 Conclusions

Sentence pairs from a general-domain corpus that seem similar to an in-domain corpus may not actually represent the same distribution of language, as measured by language model perplexity. Nonetheless, we have shown that relatively tiny amounts of this *pseudo in-domain* data can prove more useful than the entire general-domain corpus for the purposes of domain-targeted translation tasks.

This paper has also explored three simple yet effective methods for extracting these pseudo in-domain sentences from a general-domain corpus. A translation model trained on any of these subcorpora can be comparable – or substantially better – than a translation system trained on the entire corpus.

In particular, the new bilingual Moore-Lewis method, which is specifically tailored to the machine translation scenario, is shown to be more efficient and stable for MT domain adaptation. Translation models trained on data selected in this way consistently outperformed the general-domain baseline while using as few as 35k out of 12 million sentences. This fast and simple technique for discarding over 99% of the general-domain training corpus resulted in an increase of 1.8 BLEU points.

We have also shown in passing that the linear interpolation of translation models may work less well for translation model adaptation than the multiple paths decoding technique of (Birch et al., 2007). These approaches of data selection and model combination can be stacked, resulting in a compact, two phrase-table, translation system trained on 1% of the available data that again outperforms a state-of-the-art translation system trained on all the data.

Besides improving translation performance, this work also provides a way to mine very large corpora in a computationally-limited environment, such as on an ordinary computer or perhaps a mobile device. The maximum size of a useful general-domain corpus is now limited only by the availability of data, rather than by how large a translation model can be fit into memory at once.

References

Amittai Axelrod. 2006. Factored Language Models for Statistical Machine Translation. *M.Sc. Thesis. University of Edinburgh, Scotland.*

Alexandra Birch, Miles Osborne and Philipp Koehn. 2007. CCG Supertags in Factored Translation Models. *Workshop on Statistical Machine Translation, Association for Computational Linguistics.*

Stanley Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *Technical Report 10-98, Computer Science Group, Harvard University.*

Matthias Eck, Stephan Vogel, and Alex Waibel. 2004. Language Model Adaptation for Statistical Machine

Translation based on Information Retrieval. *Language Resources and Evaluation*.

George Foster and Roland Kuhn. 2007. Mixture-Model Adaptation for SMT. *Workshop on Statistical Machine Translation, Association for Computational Linguistics*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instatnce Weighting for Domain Adaptation in Statistical Machine Translation. *Empirical Methods in Natural Language Processing*.

Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. 2002. Toward a Unified Approach to Statistical Language Modeling for Chinese. *ACM Transactions on Asian Language Information Processing*.

Xiaodong He. 2007. Using Word-Dependent Transition Models in HMM-based Word Alignment for Statistical Machine Translation. *Workshop on Statistical Machine Translation, 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. 2003. Moses: Open Source Toolkit for Statistical Machine Translation. *Demo Session, Association for Computational Linguistics*.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. *Workshop on Statistical Machine Translation, Association for Computational Linguistics*.

Yajuan Lü, Jin Huang and Qun Liu. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Spyros Matsoukas, Antti-Veikko Rosti, Bing Zhang. 2009. Discriminative Corpus Weight Estimation for Machine Translation. *Empirical Methods in Natural Language Processing*.

Robert Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. *Association for Computational Linguistics*.

Preslav Nakov. 2008. Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing. *Workshop on Statistical Machine Translation, Association for Computational Linguistics*.

Franz Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Association for Computational Linguistics*

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. *Spoken Language Processing*.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, Eiichiro Sumita. 2008. Method of Selecting Training Data to Build a Compact and Efficient Translation Model. *International Joint Conference on Natural Language Processing*.