Alignment Inference and Bayesian Adaptation for Machine Translation

Kevin Duh and **Katsuhito Sudoh** and **Tomoharu Iwata** and **Hajime Tsukada**
NTT Communication Science Laboratories
2-4 Hikari-dai, Seika-cho, Kyoto 619-0237, JAPAN
{kevin.duh}@lab.ntt.co.jp

Abstract

We propose a *flexible and efficient* domain adaptation method that yields consistent improvements in machine translation (for 11 language pairs). The idea is to decompose the word alignment process into two steps, model training and alignment inference, and perform Bayesian adaptation on the latter. This modularity allows one to incorporate out-of-domain data without the need to modify existing training algorithms. We show how ideas in *sequential Bayesian methods* can be naturally applied to the word alignment problem and demonstrate various positive results on EMEA and NIST datasets.

1 Introduction

Progress in statistical machine translation (SMT) is driven both by invention of new models and by increases in training data. To date, the largest training data are bitexts from international organizations such as the United Nations and European Parliament. Although these training data are valuable, they may be from a different domain than the one where the machine translation system will be deployed. Thus a natural question is whether these so-called out-of-domain data can be exploited for improving SMT for other target domains. This is a domain adaptation problem.

In this paper, we are concerned with the common domain adaptation scenario where we have at our disposal (1) a large out-of-domain bitext, and (2) a small-to-medium sized in-domain bitext. For concreteness, let us say that our out-of-domain data is parliament bitext, and our target in-domain application is the medical domain. The in-domain bitext may be sufficient to build a reasonable medical translation system; however, our goal is to further improve upon this in-domain baseline using the large out-of-domain bitext. We will focus our attention on translation model adaptation, and use standard methods for language model adaptation.[1]

To begin addressing this problem, we divide the training pipeline for building a translation model into the following four steps:

- **Step 1: Word alignment model training:** Given bitext aligned at the sentence level, train a word alignment model.
- **Step 2: Alignment inference on bitext:** Given an alignment model, predict the alignment points on a bitext. The bitext used in Step 1 and Step 2 are usually the same, though this need not be the case (as in our proposed method).
- **Step 3: Phrase/rule extraction on bitext:** Given alignment points, find all consistent phrase pairs or translation rules.
- **Step 4: Phrase/rule score estimation:** Assign (probability) scores to the extracted rules

We ask the question: Which step should we spend our efforts? Existing domain adaptation methods

[1] Recall we can divide a SMT system into the translation model $p(e|f)$ and language model $p(e)$ components. Given a foreign sentence f, the translation e that achieves high probability in both models is preferred. Both components should be adapted under domain adaptation scenarios.

differ in where they inject out-of-domain information in the pipeline. For example, Wu et. al. (2005) improves alignment model training by interpolation with out-of-domain models (Step 1); Marton et. al. (2009) finds new phrases from additional monolingual corpora to reduce out-of-vocabulary rate (Step 3); Foster et. al. (2010) improves phrase scores by discriminative weighting (Step 4).

To the best of our knowledge, there is no prior research on alignment inference adaptation (Step 2), so we focus on it here. The advantage of focusing on Step 2 is the *flexibility*: our method does not need to modify any word alignment training algorithm, nor are we limited to particular model formalisms (e.g. phrase vs. hierarchical rules) required for the extraction/scoring steps. All we need are the standard toolsets for the training pipeline, and a pre-existing word alignment model that can generate n-best lists. Therefore, adaptation in Step 2 has wide applicability.

Although Step 1 and Step 2 are sometimes considered as a single process, we will show that the *decomposition* into two steps is quite beneficial and opens up new possibilities. Virtually all SMT systems can be interpreted as consisting of the aforementioned 4-stage pipeline (e.g. phrase-based (Koehn et al., 2003; Och and Ney, 2004), hierarchical (Chiang, 2007; Wu, 1997), and tree-based (Quirk et al., 2005; Galley et al., 2004; Mi et al., 2008)).

Our approach can be briefly summarized as follows: First, we train a word alignment model on a large general-domain dataset, then predict the alignment points for an in-domain bitext. The n-best list of predictions are used to compute a Bayesian prior indicating the *a priori* belief of any two words being aligned. Then, alignment inference of in-domain bitext is viewed as a *sequential Bayesian update* on weighted alignment matrices. The idea is to effectively balance the uncertainty in alignment from both in-domain and general-domain bitexts.

The contribution of this paper is two-fold:

- We identify *alignment inference* as an open area for SMT adaptation research.
- We propose a method that models alignment inference as Bayesian adaptation of alignment matrices, which is effective on various datasets.

In the following, Section 2 describes our proposed Bayesian adaptation method. Section 3 discusses experimental results from two tasks (EMEA, NIST) and 11 language pairs. Section 4 reviews related work, and Section 5 discusses our conclusions.

2 Alignment Inference Adaptation

2.1 General Bayesian Framework

Alignment inference is the task of predicting alignment points on a given sentence-pair, using a pre-trained word alignment model. Let (e_1^I, f_1^J) be a sentence-pair consisting of I English words $\{e_1, e_2, \ldots, e_I\}$ and J Foreign words $\{f_1, f_2, \ldots, f_J\}$. We define an alignment matrix $\mathbf{A} \in \{0,1\}^{I \times J}$ to be an I-by-J matrix where each element A_{ij} indicates whether words e_i and f_j is aligned ($A_{ij} = 1$) or not ($A_{ij} = 0$).

We already have a pre-trained word alignment model, such as IBM Model 4. Suppose this model is trained only on in-domain bitext, so we call it M^{in}. The goal of alignment inference, under the Bayesian framework, is to compute the posterior $P(\mathbf{A}|M^{in}; e_1^I, f_1^J)$ for the sentence pair (e_1^I, f_1^J). This is obtained by Bayes theorem:

$$p(\mathbf{A}|M^{in}; e_1^I, f_1^J) \propto p(M^{in}|\mathbf{A}; e_1^I, f_1^J) p(\mathbf{A}; e_1^I, f_1^J) \quad (1)$$

where $l^{in}(\mathbf{A}) \equiv p(M^{in}|\mathbf{A}; e_1^I, f_1^J)$ is the likelihood of an alignment result under the model M^{in} and $p(\mathbf{A}; e_1^I, f_1^J)$ is the prior over alignments. In other words, the optimal alignment should be both highly likely according to the in-domain model M^{in}, *and* have high probability *a priori*. The prior probability is gleaned from large general-domain data, details of which will be discussed in the next sections.

For tractable computation of Equation 1, we assume that the probability of an alignment matrix $\mathbf{A}$ can be decomposed into a product of its matrix elements. Equation 1 can now be rewritten as:[2]

$$p(\mathbf{A}|M^{in}) \propto \Pi_{1 \le i \le I, 1 \le j \le J}\, p(A_{ij}|M^{in}) \quad (2)$$

[2]To simplify notation, we have now dropped the conditioning terms e_1^I, f_1^J but remember that inference is always done for a particular sentence pair.

Algorithm 1 Alignment Inference by Sequential Bayesian Adaptation

Input: Bitext: T^{in}, T^{out}

Output: Posterior alignment matrix $\mathbf{A}$

for each sentence-pair in T^{in}

M^{in} = AlignModelTrain(T^{in})

M^{gen} = AlignModelTrain(concat[T^{in} + T^{out}])

for each sentence-pair in T^{in} **do**

  for each $(e_i, f_j), 1 \leq i \leq I, 1 \leq j \leq J$ **do**

    Estimate likelihood a_{ij}, b_{ij} by Eq. 3 and 4.

    Estimate prior α_{ij} and β_{ij} by Eq. 7 and 8.

    Set posterior of A_{ij} as $\frac{a_{ij}+\alpha_{ij}}{a_{ij}+b_{ij}+\alpha_{ij}+\beta_{ij}}$ (Eq. 6)

  end for

  Recreate alignment matrix $\mathbf{A}$ by Eq. 2

end for

Each of the individual terms in Equation 2 is naturally modeled by a Bernoulli-Beta distribution, as A_{ij} is a binary variable. The advantage of this decomposition is we can perform Bayesian updates on Bernoulli-Beta in closed-form, which is scalable for training on datasets with millions of words.

Specifically, the probability that e_i and f_j are aligned follows a Bernoulli distribution with parameter μ: $P(A_{ij}|\mu) = \mu^{A_{ij}}(1-\mu)^{(1-A_{ij})}$. If we were a non-Bayesian (frequentist), we would estimate a single value for μ using the likelihoods of N-best alignments resulting in $A_{ij} = 1$ vs. $A_{ij} = 0$:

$$a_{ij} = \sum_{\mathbf{A}' \in N(e_1^I, f_1^J)} l^{in}(\mathbf{A}')\delta(A_{ij} = 1)/Z \quad (3)$$

$$b_{ij} = \sum_{\mathbf{A}' \in N(e_1^I, f_1^J)} l^{in}(\mathbf{A}')\delta(A_{ij} = 0)/Z \quad (4)$$

$$P(A_{ij} = 1|\hat{\mu}) = \hat{\mu} = \frac{a_{ij}}{a_{ij} + b_{ij}} \quad (5)$$

Here $N(e_1^I, f_1^J)$ represents the N-best list of alignments, $\delta(\cdot)$ is the identity function, and $Z = \sum_{\mathbf{A}' \in N(e_1^I, f_1^J)} l^{in}(\mathbf{A}')$ is a normalizer . So predicted alignment probability is simply $\frac{a_{ij}}{a_{ij}+b_{ij}}$, the percentage of times e_i and f_j are aligned in the N-best list (Section 2.2 provides details).

On the other hand, being Bayesian, we do not set μ to a single value but allow it to be a random variable, following the beta (conjugate) prior: $P(\mu|\alpha_{ij}, \beta_{ij}) = \frac{\Gamma(\alpha_{ij}+\beta_{ij})}{\Gamma(\alpha_{ij})\Gamma(\beta_{ij})}\mu^{\alpha_{ij}-1}(1-\mu)^{\beta_{ij}-1}$. The Gamma function $\Gamma(\cdot)$ serves as a normalization constant and α_{ij} and β_{ij} are hyperparameter estimated from general-domain data. After observing $l^{in}(A_{ij})$, the posterior of μ is: $P(\mu|M^{in}, \alpha_{ij}, \beta_{ij}) = \frac{\Gamma(\alpha_{ij}+a_{ij}+\beta_{ij}+b_{ij})}{\Gamma(\alpha_{ij}+a_{ij})\Gamma(\beta_{ij}+b_{ij})}\mu^{\alpha_{ij}+a_{ij}-1}(1-\mu)^{\beta_{ij}+b_{ij}-1}$. Finally, the posterior alignment can be calculated by integrating out μ, which leads to a simple formula for inference[3]:

$$\begin{aligned} & p(A_{ij} = 1|M^{in}, \alpha_{ij}, \beta_{ij}) \\ = & \int_0^1 p(A_{ij} = 1|\mu)p(\mu|M^{in}, \alpha_{ij}, \beta_{ij})d\mu \\ = & \frac{a_{ij} + \alpha_{ij}}{a_{ij} + b_{ij} + \alpha_{ij} + \beta_{ij}} \end{aligned} \quad (6)$$

Compared to frequentist estimation (Equation 5), Bayesian inference allows the incorporation of additional "counts" α_{ij} and β_{ij}. Intuitively, α_{ij} represents the prior belief that e_i and f_j are aligned, and β_{ij} represents the opposing belief that they are not.

To summarize: looking back to the original Eq. 1, we see that the posterior alignment matrix is now computed element-wise by Eq. 6. The next subsections detail how the Eq. 6 is computed in practice. Note that while we have introduced a method that achieves a nice balance between Bayesian theory and practical efficiency, it is by no means the only way to perform alignment inference adaptation.

[3]For an introduction of sequential Bayesian update and the derivation for this particular form, please see (Bishop, 2006).

2.2 Calculation of Likelihood

The likelihood terms (a_{ij} and b_{ij}) in Eq. 6 can be computed effectively using the algorithm proposed by Liu et. al. (2009): First, we extract N-best alignments of sentence pair (e_1^I, f_1^J) from M^{in} in forward and reverse directions. Each alignment solution in the N-best list is an alignment matrix $\mathbf{A}'$, with likelihood score $l^{in}(\mathbf{A}')$. To merge the alignments in both directions, we take the $N \times N$ cross-product of N-best lists and multiply their likelihood scores. Finally, for each element in the alignment matrix A_{ij}, we compute a_{ij} and b_{ij} by Eq. 3 and 4.

Note that if our word alignment model readily outputs alignment posteriors (e.g. IBM model), we can readily obtain a_{ij} and b_{ij} as alignment posterior for words e_i, f_j. We opt for Liu's (2009) approach here since it makes our overall adaptation approach more flexible to different choices of word aligners (e.g. discriminative aligners that do not output posteriors). In practice, 100-best list appears sufficient to approximate the posterior.

2.3 Calculation of Prior

The hyperparameters α_{ij} and β_{ij} represent prior knowledge from general-domain data. We have been using the terms "general-domain" and "out-of-domain" somewhat interchangeably until this point. Now we will define it clearly: general-domain bitext T^{gen} is the *concatenation* of in-domain bitext T^{in} and all available out-of-domain bitexts T^{out}, i.e. simply all the data for a language pair. We will use general-domain bitext (not out-of-domain) to estimate hyperparameters because we believe that the prior ought to express general beliefs about a language pair. This distinction is subtle but important: we are adapting from general- to-specific domain, rather than across (out-to-in) domains.

We define the prior terms (α_{ij} and β_{ij}) as alignment probabilities under a general model M^{gen}. M^{gen} is trained on a concatenation of in-domain and out-of-domain bitexts to capture general language characteristics. This model then generates n-best alignments on the *in-domain* portion of the bitext. Similar to the likelihood, the prior is estimated as:

$$\alpha_{ij} = \sum_{\mathbf{A}' \in N_{gen}(e_1^I, f_1^J)} l^{gen}(\mathbf{A}')\delta(A_{ij} = 1)/Z \quad (7)$$

This is similar to Eq. 3, except we use scores from the general model M^{gen} rather than M^{in}; β_{ij} is calculated analogously to Eq. 4:

$$\beta_{ij} = \sum_{\mathbf{A}' \in N_{gen}(e_1^I, f_1^J)} l^{gen}(\mathbf{A}')\delta(A_{ij} = 0)/Z \quad (8)$$

2.4 Summary and Caveat

The pseudocode for the overall algorithm is presented in Algorithm 1. Basically, the alignment matrix posteriors are computed for each sentence pair by combining statistics from in-domain and general-domain models.

It is worth noting two caveats:

- Our Bayesian view is that there is a prior alignment matrix, which is updated by in-domain model statistics. This differs from previous work in Step 1, e.g. (Wu et al., 2005), which adopts a prior for alignment *model parameters*. The distinction between adapting inference results and model parameters is an important one, and this is what gives us a flexible general-purpose method.

- Eq. 6 does not contain a tuning parameter between likelihood a_{ij} and prior α_{ij}. This arises from the *sequential* Bayesian update perspective, where each additional sample is counted equally. It may be beneficial to have a parameter if it can be tuned well without of overfitting, but we do not consider it here.

3 Experiments

3.1 Datasets and Setup

We evaluate our proposed method under two tasks: The EMEA task involves the translation of medical texts from the European Medical Agency (Tiedemann, 2009). We test on ten language pairs–Danish (da), German (de), Greek (el), Spanish (es), Finnish (fi), French (fr), Italian (it), Dutch (nl), Portuguese (pt), and Swedish (sv)–all translating into English. The out-of-domain data are parliamentary texts from Europarl (Koehn, 2005).

The NIST task involves translating newswire text using Chinese-to-English NIST OpenMT 2008 data.

	EMEA										NIST	
	da	de	el	es	fi	fr	it	nl	pt	sv	mt06	mt08
in-domain	45.3	35.5	41.3	45.0	33.6	46.8	47.7	45.6	46.3	45.3	27.7	24.4
general	46.1	36.1	41.6	**46.9**	34.0	47.8	49.2	46.1	47.0	45.2	**28.7**	24.6
bayes	**47.1***	**36.4***	**42.5***	46.2	**34.6***	**47.9**	**49.3***	**46.2***	**47.5***	**45.9***	**28.7**	**25.0***

Table 2: Main Results: Test BLEU for EMEA (da,de,...,sv) and NIST (mt06,08): Best results are in bold-font. Statistical significant improvement over **general-domain model** is indicated by asterisk (*).

	EMEA	NIST
Language Pair	10 European	zh-en
In-domain	Medicine	Newswire
#sent train	100k	250k
#word train	1.2M	5.6M
#sent devset	2k	2.4k (MT04,05)
#sent testset1	2k	616 (MT06)
#sent testset2	-	691 (MT08)
Out-of-domain	Parliament	Heterogeneous
#sentence	1.2M	4.8M
#word	25M	107M

Table 1: Dataset statistics

We select a subset of newswire text from the allowed resources lists[4] as in-domain data. Out-of-domain data is heterogeneous, consisting of Hansards, broadcast conversations, weblogs, etc. The data statistics are shown in Table 1.

We compare 3 phrasal SMT systems:

- **in-domain model:** Step1: Train word alignment model on in-domain bitext (M^{in}). Step 2-to-4: Alignment inference on in-domain text, followed by phrase extraction and scoring.
- **general-domain model:** Step1: Train word alignment model on concatenated in-domain and out-of-domain bitext (M^{gen}). Step2-to-4: same as **in-domain model**. This simple approach is a strong adaptation baseline competitive in many tasks, c.f. (Duh et al., 2010).
- **bayes:** Step1-to-2: Algorithm 1. Step3-to-4: same as **in-domain model**.

Note that out-of-domain information is used only up to step 2 and excluded in further steps of the pipeline. This clarifies the analysis: if we were to include out-of-domain bitext for phrase extraction, our SMT system might acquire new phrases, which reduces out-of-vocabulary rate and confounds the analysis of alignment inference results. Further, from preliminary experiments, we found that adding out-of-domain bitext to all four steps actually *degrade* results sometimes, due to increased ambiguity of additional translation options on the target side.

For all systems, we use the Moses decoder, adapted SRILM 3gram (EMEA) and 4gram (NIST), MERT for weight optimization, and GIZA++ (Model4) as the underlying word alignment training tool. Phrase tables are extracted from alignment matrices using the method of (Liu et al., 2009).

3.2 Main Results

Table 2 summarizes all our main results. We see that **bayes** gives robust improvements in testset BLEU. For example, for Danish-to-English translation, using in-domain data by itself achieves 45.3 BLEU (**in-domain**). This can be improved to 46.1 BLEU by concating out-of-domain data (**general**). The proposed method, however, further improves the result to 47.1 BLEU (**bayes**). For the NIST dataset, we see that **bayes** improves upon **general** and **in-domain** for the MT08 testset, and ties with **general** for the MT06 testset.

On average, **bayes** improves over **in-domain model** by 1.1 BLEU points. Further, in 9 of 12 cases, **bayes** also outperforms **general-domain model** by statistically significant margins, p <0.05 (Zhang et al., 2004). We thus conclude that the proposed method is robust under adaptation scenarios.

3.3 Analyses of Alignments

We are also interested in checking if BLEU improvements correlate with quantifiable alignment

[4]www.itl.nist.gov/iad/mig/tests/mt/doc/

improvements. This evaluation is possible since the NIST dataset contains some manual alignment annotations (LDC2006E93). We identified 892 sentence-pairs in our in-domain bitext that have manual alignments. Note that this supervised information is never used in any part of our method.

Figure 1 shows alignment precision/recall. The curve is computed by thresholding the estimated weighted aligment matrix at different levels and computing precision and recall with the gold reference. Interestingly, **bayes** performs best, but **general-domain** also improves alignment. We conjecture this is a common phenomenon, since many words have the same translation *regardless* of domain differences. So using out-of-domain data for alignment (and not for finding new phrases) is relatively robust.

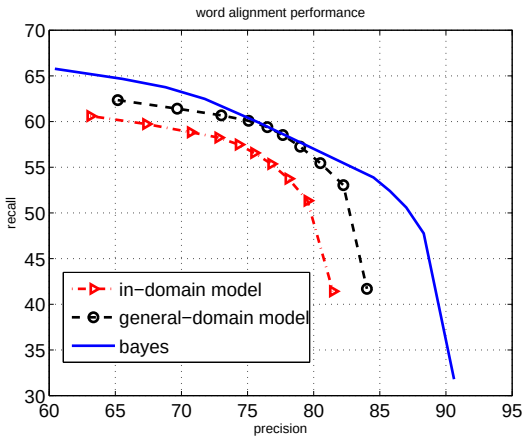

Figure 1: Alignment Precision/Recall Curve using 892 manually-annotated sentence-pairs in NIST task.

4 Related Work

There is a wide variety of previous work in SMT domain adaptation. It might be helpful to categorize the methods based on (a) whether the method focuses on a particular step in the training pipeline, and (b) if so, which step. Following the training pipeline in Section 1, we briefly survey the approaches in prior work:

1) Word alignment model training: Assume a probabilistic model of alignment, where the parameters (e.g. lexical translation probabilities in the IBM Models) are estimated from a mix of in-domain and out-of-domain data. One method is to interpolate separate sets of parameters estimated from in-domain and out-domain data. Wu et. al.(2005) sets the interpolation weights to be proportional to the relative frequency of observances in in-domain and out-of-domain data, while (Civera and Juan, 2007) treats it as a hidden parameter in a mixture model. These methods are similar to ours in that the motivation is to improve alignments, but differs in that the focus is on training (not inference).

2) Alignment inference: To the best of our knowledge, there is no previous work in this area. The model training of Step 1 is related but not the same. Instead, alignment combination works (Deng and Zhou, 2009) may give some insights.

3) Phrase extraction: Out-of-domain text may contain unseen phrases useful for in-domain data. One approach attempts to discover paraphrases from large monolingual corpora (Marton et al., 2009; Snover et al., 2008). Another is a self-training approach that translates source in-domain text and re-trains the translation model on synthetic data (Bertoldi and Federico, 2009; Ueffing et al., 2007).

4) Scoring: Adaptation in the scoring step is the most direct way to improve results since it is the step closest to the final translation model. In fact, one could argue that all the previous steps are simply pre-processing to narrow down the size of ruleset/phrasetable; if scores are well-tuned, good translations can be achieved even if the ruleset is infinite in size. Recent approaches to score adaptation involve combining in-domain and out-of-domain scores at either the sentence or the phrase level (Shah et al., 2010; Matsoukas et al., 2009; Foster et al., 2010). A promising aspect about the latter two papers, in particular, is that they are able to incorporate *supervised* information (likelihood or expected TER on the dev set) for score adaptation.

We emphasize that our contribution is orthogonal to previous work: alignment inference adaptation can be combined with any adaptation method in other parts of the pipeline. It remains to be seen whether the improvements are additive: while our results are positive in both alignment and final translation performance, some work have shown weak correlation between the two (Ayan and Dorr, 2006; Fraser and Marcu, 2009).

There are also adaptation methods that do not tar-

get a particular step in the training pipeline. For example, the information retrieval approach (Hildebrand et al., 2005) begins by identifying a subset of out-of-domain bitext most similar to in-domain; this data subset can be used for any (or all) steps of the training pipeline. An alternative approach is to train separate translation models for in-domain and out-of-domain data, then combine the final models log-linearly (Koehn and Schroeder, 2007) or dynamically (Finch and Sumita, 2008; Lü et al., 2007).

5 Conclusions and Future Work

We proposed a flexible and efficient method for domain adaptation in machine translation. The idea is to decompose the word alignment process into model training and alignment inference, and view the latter as a sequential Bayesian update problem. The advantages of our approach are:

1. Its modularity enables the use of any model training algorithm for word alignment, as long as it outputs N-best lists or posteriors.

2. It gives consistent improvements over a multitude of datasets (2 tasks and 11 language pairs).

We have shown how alignment inference can be efficiently modeled in a Bayesian way by using Bernoulli-Beta distributions. One direction of future work is to relax the independence assuption used in Eq. 2. For example, we might capture dependencies among alignment points as a 2-D Markov Random Field and develop tractable variational or MCMC inference algorithms to compute the posterior. Another direction of future work is to explore alternative non-Bayesian methods for combining alignment inference results, such as dual decomposition (DeNero and Macherey, 2011). Finally, it would be interesting to compare and combine the alignment inference results with methods that directly adapt the model parameters (e.g. (Wu et al., 2005)).

References

Necip Ayan and Bonnie Dorr. 2006. Going beyond AER: an extensive analysis of word alignments and their impact on MT. In *ACL*.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *WMT*.

Chris Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).

Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *WMT*.

John DeNero and Klaus Macherey. 2011. Model-based aligner combination using dual decomposition. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Yonggang Deng and Bowen Zhou. 2009. Optimizing word alignment combination for phrase table training. In *ACL-IJCNLP (short papers)*.

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) - Technical Papers Track*.

Andrew Finch and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *WMT*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*.

Alexander Fraser and Daniel Marcu. 2009. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL*.

Almut Silja Hildebrand, Matthias Eck, and Stephan Vogel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *EAMT*.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *WMT*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT*.

Philipp Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *MT Summmit*.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *EMNLP*.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *EMNLP-CoNLL*.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *EMNLP*.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *EMNLP*.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *ACL*.

Franz Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency tree translation, syntactically informed phrasal smt. In *ACL*.

Kashif Shah, Loic Barrault, and Holger Schwenk. 2010. Translation model adaptation by resampling. In *WMT*.

Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *EMNLP*.

Jörg Tiedemann. 2009. News from opus - a collection of multilingual parallel corpora with tools and interface. In *RANLP*.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Hua Wu, Haifeng Wang, and Zhanyi Liu. 2005. Alignment model adaptation for domain specific word alignment. In *ACL*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3).

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *LREC*.