Translating Structured Documents

George Foster **Pierre Isabelle** **Roland Kuhn**

National Research Council Canada
first.last@nrc.gc.ca

Abstract

Machine Translation traditionally treats documents as sets of independent sentences. In many genres, however, documents are highly structured, and their structure contains information that can be used to improve translation quality. We present a preliminary approach to document translation that uses structural features to modify the behaviour of a language model, at sentence-level granularity. To our knowledge, this is the first attempt to incorporate structural information into statistical MT. In experiments on structured English/French documents from the Hansard corpus, we demonstrate small but statistically significant improvements.

1 Introduction

It is a standard assumption in statistical machine translation (SMT) that sentences within a document can be translated independently. This is clearly an approximation, since context is often helpful, and occasionally essential, for correct translation. One obvious dependence on context is for the resolution of anaphora, as in the sentence *He sees it*, where the referent of *it* must be known in order to translate into languages with gender-marked pronouns such as French. A more subtle dependence on context is a sentence's role within the document. For instance, it may occur in an introductory paragraph or in the main body of a text; it may be a quotation or a line spoken by a character in a work of fiction; or it may play a particular rhetorical role within a monograph. All these factors can influence translation, and hence they are potentially useful sources of information for SMT.

In this paper, we explore the idea of using document context to improve SMT. In general, this means exploiting information that is available in a document's structure, but that is typically discarded when the document is transformed into a sequence of sentences for automatic translation. The document structure may be specified by markup, or it may be inferrable from formatting instructions or other information. Most texts, even relatively unstructured ones such as email or blogs, have some degree of heterogeneity that can in principle be exploited to improve translation—although how best to accomplish this, and how productive a strategy it will be, clearly depends on genre.

Our approach is to characterize each sentence in a document with a vector of feature values derived from the document's structure, then translate it with a model optimized for those values. In its use of sentence-level information, our method is related to information-retrieval inspired approaches to domain adaptation (Zhao et al., 2004; Hildebrand et al., 2005; Lü et al., 2007), which seek parts of the training corpus that resemble sentences in the current document. However, those approaches pool matches across sentences, and build a single adapted model for each source document, rather than (potentially) one for each sentence. Our method is also different from previous work that relies on intrinsic surface cues to categorize sentences for translation according to whether they are interrogative or declarative (Finch and Sumita, 2008), or match an ad hoc class (Hasan and Ney, 2005). We rely on extrinsic prop-

erties derived from the whole document, and also, unlike all previous work we are aware of, characterize sentences along multiple axes (eg, a sentence might be attributed to a particular speaker *and* belong to a specific section within a document). Finally, unlike anaphora resolution and discourse analysis applied to translation (Marcu et al., 2000), our method does not explicitly depend on the content of sentences other than the current one.

We formalize our notion of document translation in section 2, and present two alternative modeling techniques. Section 3 describes English/French translation experiments with a version of the Canadian Hansard corpus that has rich structural information encoded in XML markup. Related work is discussed in section 4, and section 5 concludes and suggests some possibilities for future work.

2 Document Translation

SMT seeks the translation hypothesis $\hat{t}$ that has highest probability according to a model conditioned solely on the current source sentence s:

$$\hat{t} = \underset{t}{\operatorname{argmax}} \log p(t|s).$$

Assuming that information about the source sentence's document context is captured in a feature vector d, our document translation approach is described by:

$$\hat{t} = \underset{t}{\operatorname{argmax}} \log p(t|s,d), \quad (1)$$

where d will typically consist of one or more discrete features such as those in the examples in section 1.

To model (1), we suppose that a training corpus in which each sentence pair is tagged with document features is available. (Obviously, this will apply only in cases where the training and testing domains are identical or closely related.) Given the typical SMT model structure, there are three main sites for incorporating conditioning on d: the top-level log-linear model[1], the language model (LM), and the translation model (TM). The log-linear model is not ideal for directly capturing dependence on d, since it is trained on a small development set of approximately 1000 sentences: even using a MIRA-like algorithm (Chiang et al., 2009), and assuming simple features that connect target words to d, one would have to carefully select only a very small subset of all potential features. The language and translation models can use much larger feature sets, but they also face a sparsity problem in that the number of training examples available for a particular vector d can be arbitrarily small. This is especially severe for the translation model, which lacks the powerful backoff-based smoothing algorithms used in langauge modeling, and which usually depends on an IBM-model training process that degrades badly on small data. For this preliminary effort, we therefore concentrated on the language model.

As just mentioned, the challenge in constructing a language model $p(w|h,d)$, where h is an ngram context for word w, is data sparsity due to the conditioning on d. The normal method for dealing with sparsity in h—backing off to shorter contexts (Goodman, 2001)—cannot be applied directly because there is no natural back-off ordering for the features in d. Methods for handling similar situations have been devised in a factored model setting (Bilmes and Kirchhoff, 2003), but these lack straightforward training procedures. We opted instead for two simple solutions: splitting d into its component features d_i and training a specific model for each; and clustering rare vectors d together to increase the amount of data available for each.

2.1 Feature-Specific Models

The training procedure for feature-specific models is extremely simple. For each feature d_i in d:

1. Partition the target half of the training corpus into sets of sentences characterized by each different value that d_i can take on.

2. Train an LM on each corpus partition.

This yields models $p(w|h,d_{ij})$, one for each jth value of each ith feature. These could be used directly for translating test-set sentences characterized by d_{ij}, but there is still no guarantee that d_{ij} occurs often enough in the training corpus to produce an LM that generalizes well. We therefore smooth using a word-level mixture with a global LM:

$$p_s(w|h,d_{ij}) = \alpha_{ij} p(w|h,d_{ij}) + (1-\alpha_{ij}) p(w|h). \quad (2)$$

[1] That is, $\log p(t|s,d)$, assumed to be a weighted linear combination of features that can be interpreted as log probabilities.

To set mixture weights α_{ij}, we used a dynamic smoothing technique similar to dynamic LM domain adaptation (Foster and Kuhn, 2007). First, steps 1 and 2 above are repeated on the *source* half of the training corpus to produce a set of source-language LMs $p'(w|h, d_{ij})$ that correspond one to one with their target-language counterparts. Then, for each d_{ij} that occurs in the current source document to be translated, a mixture weight α'_{ij} is learned using the EM algorithm to maximize the probability of the source sentences tagged with d_{ij}, according to the source-language counterpart of (2).[2] Finally, the source-side weights are simply transferred to the target side ($\alpha'_{ij} \rightarrow \alpha_{ij}$) and used in (2). Although this procedure lacks compelling theoretical justification, it works well in practice (α'_{ij} tends to approximate α_{ij} well), and it allows the model to closely reflect the properties of the current source document.

The one-versus-all mixture in (2) is appropriate for features with small numbers of values, such as the major logical divisions of a document. However, features such as speaker identity that can take on many values will partition the corpus finely, and may benefit from being able to share information across partitions. As this does not occur with the all-versus-one approach, we also tried mixing over all values of a given feature (ie, over all LMs learned from the training corpus for that feature):

$$p_s(w|h, d_{ij}) = \sum_k \alpha_{ijk}\, p(w|h, d_{ik}), \qquad (3)$$

where $\sum_k \alpha_{ijk} = 1$, and for notational convenience we designate the global model—included in the mixture—as $p(w|h, d_{ij0})$. The weights α_{ijk} were learned using the dynamic smoothing procedure above. To counter overfitting due to the large number of parameters, we incorporated MAP smoothing into the EM procedure, following (He, 2007). This uses a modified M-step update: $p(x) = [c(x) + \lambda p_0(x)]/(\sum_x c(x) + \lambda)$, where $p(x)$ is a mixture-component probability, $c(x)$ is a corresponding expected count, and $p_0(x)$ is a prior probability for which we used weights α_i learned by pooling all values of feature d_i in the current source document. The prior weight λ was set to 10, based on preliminary experiments with a development set.[3]

The procedure we have just described constructs a family of smoothed language models $p_s(w|h, d_{ij})$ for feature d_i that is specific to the current source document D. To decode a sentence from D, we use the value d_{ij} that d_i takes on for that sentence to select the appropriate member of the family. We tried two ways of combining model families from different document features d_i: a log-linear combination, with model weights set by MERT (Och, 2003); and a linear combination, with weights set to maximize the likelihood of a target-language development corpus. Both combinations involve one weight per document feature d_i.

2.2 Clustered Models

A weakness of the approach outlined in the previous section is that it implicitly assumes features are independent. Clearly this will not always be the case. Furthermore, many feature vectors occur often enough in the training corpus to allow reliable LMs to be produced. To capitalize on this, we used a simple clustering method that attempts to group low-frequency vectors together to increase the reliability of the resulting LMs, while preserving the data associated with high-frequency vectors. The algorithm is as follows:

1. Create an initial clustering by grouping all target sentences having identical feature vectors together.
2. Combine the cluster having the smallest number of tokens with the most similar other cluster.
3. Stop if all clusters have token counts greater than or equal to a threshold f, otherwise repeat from step 2.

Note that this is a hard agglomerative algorithm that produces a hierarchical structure which can be used to assign any original feature vector associated with a "leaf" cluster to the appropriate final cluster.

To determine cluster similarity, we relied solely on the contents of the clusters, rather than their associated feature vectors. Since our aim is to be able

[2] For values of d_{ij} that don't occur in the training corpus, α'_{ij} is set to zero.

[3] For efficiency, since our implementation computes the mixture in (3) dynamically during decoding, we limited the number of components to the 20 having highest weights α_{ijk}.

to train informative LMs on each final cluster, we would like the clusters to be as homogeneous as possible from the perspective of an ngram LM. To accomplish this, we merged clusters that resulted in the lowest drop in corpus likelihood, defined as the probability assigned to a cluster by an LM trained on that cluster. For two merge candidates C_1 and C_2, this merge cost is:

$$\text{cost}(C_1, C_2) = \log[p(C_1)p(C_2)] - \log p(C_1 \cup C_2)$$

where $p(C)$ is the probability assigned to cluster C by a language model trained on C. For efficiency, we used unigram probability to approximate higher-order ngram LMs. Under this assumption, it is easy to show that merge cost can be calculated using the following formula that requires iteration only over words in the intersection of C_1 and C_2:

$$\text{cost}(C_1, C_2) = T_1 \log \frac{T}{T_1} + T_2 \log \frac{T}{T_2} - \sum_{w \in C_1 \cap C_2} c_1(w) \log \frac{c(w)}{c_1(w)} - c_2(w) \log \frac{c(w)}{c_2(w)},$$

where $c_i(w)$ is the count of word w in C_i, $c(w) = c_1(w) + c_2(w)$, $T_i = \sum_w c_i(w)$, and $T = T_1 + T_2$.

To translate with clustered models, we first identify the correct model for each feature vector d that occurs in the current source document using the cluster hierarchy as outlined above. (Feature vectors that don't appear in the training corpus are mapped to the most similar vector that does.) Then we apply dynamic smoothing as in (2) and use the resulting smoothed models $p_s(w|h, d)$ when translating the matching sentences.

3 Experiments

We performed experiments in English/French translation (both directions) using a standard phrase-based SMT system, with a large corpus of structured Hansard documents.

3.1 Corpora and Features

The Hansard corpus consists of transcripts of Canadian parliamentary proceedings. During the manual transcription process, the text is annotated with various items of information such as the speaker, the purpose of the speech (a motion, debate, question to the government, etc.), and the language used (English or French). These annotations are encoded in XML and preserved through translation. The resulting corpus is aligned at the paragraph level (facilitating subsequent high-quality automatic sentence alignment), and each "document"—generally a single day's proceedings—contains a rich set of structural information.

We extracted five main features for each aligned sentence pair:

- *session*: The current parliamentary session. Sessions can span years, so this feature is constant for any given document. There are 8 sessions in our data, ranging from 2001 to 2009.

- *srclang*: The language used by the current speaker. Values are English (roughly 75% of the corpus) and French (25%).[4]

- *speaker*: The name of the person speaking. This takes on 586 different values in our training corpus, and follows a fairly Zipfian distribution, from *Peter Milliken* (almost 150k sentences) to *Audrey O'Brien* (4 sentences).

- *title*: Characterizes the current activity into one of 45 categories, ranging from general—eg, *Debate* (over 1m sentences) and *QuestionsAndComments* (500k sentences); to very specific—*RoyalAssent* (290 sentences) and *FirstReadingOfSenatePublicBills* (145 sentences).

- *section*: A hierarchical feature that is partially complementary to *title*, with 4 top-level distinctions pertaining to daily routine in parliament, and 95 full values, including exotic ones such as *Intervention-Content-ParaText-Poetry-Verse-Line*. Since the distribution of values is highly skewed (2.5m sentences are tagged as *Intervention-Content-ParaText*), and many lower-frequncy tags do not appear to be consistently assigned, we used only the top-level distinctions for feature-specific models (but all values for clustering).

[4] It is interesting to note that when this value does not match the current source language in our experiments, we are actually translating previously manually translated material and using the original source text as a reference.

corpus	sentences	words (en)	words (fr)
train	2.9m	60.5m	68.6m
dev	2,002	40k	45k
test1	2,148	43k	48k
test2	2,166	45k	50k

Table 1: Corpus sizes

Table 1 summarizes our training corpus. We reserved the most recent five documents (dating from December 2009) for development and testing material, and extracted the dev and test corpora shown. Since some of the original documents were much larger than typical devtest sizes, we sampled subsets of them for the dev and test sets, in such as way as to preserve the original distribution of features.

3.2 System

We used a standard one-pass phrase-based system (Koehn et al., 2003), with 7 features in the baseline configuration:

- relative-frequency TM probabilities in both directions (2 features)
- "lexical" TM probabilities in both directions (2 features)
- 4-gram LM with Kneser-Ney smoothing
- word-displacement distortion
- word count

Feature weights were set using Och's MERT algorithm (Och, 2003) to maximize dev-set BLEU score.

The training corpus was word-aligned using both HMM and IBM2 models; the phrase table consists of the union of phrases extracted from these separate alignments, with a phrase length limit of 7. It was filtered to retain the top 30 translations for each source phrase using the TM part of the current log-linear model. Lexical probabilities were estimated using the method described in (Zens and Ney, 2004).

3.3 Results

A preliminary step to generating translation results is clustering the corpus as described in section 2.2. Table 2 shows the results, for various frequency-threshold (*f*) values (for the *20n* value, clustering was run until exactly 20 clusters remained). The results are similar for clustering on the English (for translation into English) and French (for translation into French) sides of the corpus. The largest cluster size is identical for $f = 0$ (no clustering) and for $f = 1k$, indicating that it was not merged at this threshold; in fact, the largest 100 or so clusters survived intact for both English and French for $f = 1k$. However, at $f = 10k$, no original cluster survived.

f	en #	en max	fr #	fr max
0	57,428	5,167	57,428	5,167
1k	12,808	5,167	13,456	5,167
10k	3,317	10,976	3,683	7,307
100k	385	43,562	452	42,709
20n	20	351,209	20	238,292

Table 2: Clustering results. The # columns give the number of clusters, and the *max* columns give the number of sentences in the largest cluster.

method	test1	test2	avg
baseline	41.11	38.23	39.67
session 1-all	41.27	38.45	39.86
srclang 1-all	41.06	38.30	39.68
speaker 1-all	41.34	38.26	39.80
title 1-all	41.31	38.44	39.87
section 1-all	41.08	38.25	39.66
session multi	41.12	38.24	39.68
srclang multi	41.17	38.32	39.74
speaker multi	41.23	38.18	39.71
title multi	41.35	38.43	39.89
section multi	41.20	38.30	39.75
clust 1k	41.12	38.12	39.62
clust 10k	41.25	38.37	39.81
clust 100k	41.15	38.31	39.73
clust 20n	41.08	38.23	39.65
log comb 1-all	41.33	38.44	39.88
lin comb 1-all	41.34	38.45	39.89
log comb multi	41.33	38.41	39.87
lin comb multi	41.36	38.56	**39.96**

Table 3: Results for French to English translation.

Tables 3 and 4 give BLEU scores for French to English translation and the reverse. The top two blocks in each table, after the baseline, contain, respectively, results for feature-specific models with 1-versus-all smoothing mixtures (equation 2), and

method	test1	test2	avg
baseline	41.38	37.62	39.50
session 1-all	41.71	37.96	39.84
srclang 1-all	41.91	38.10	40.00
speaker 1-all	41.89	37.82	39.86
title 1-all	41.79	37.84	39.81
section 1-all	41.64	37.77	39.70
session multi	41.80	37.93	39.87
srclang multi	41.59	37.78	39.68
speaker multi	41.91	37.79	39.85
title multi	41.83	37.90	39.86
section multi	41.78	37.78	39.78
clusters.1k	41.57	37.58	39.58
clusters.10k	41.76	37.84	39.80
clusters.100k	41.77	37.93	39.85
clusters.20n	41.81	37.83	39.82
log comb 1-all	41.92	37.80	39.86
lin comb 1-all	42.02	38.03	**40.03**
log comb multi	41.86	37.79	39.83
lin comb multi	41.93	37.85	39.89

Table 4: Results for English to French translation.

multiple-feature smoothing mixtures (equation 3). The next block gives results for clustering models with various frequency thresholds. Finally, the bottom two blocks are log-linear and linear combinations of 1-versus-all and multiple-feature specific models over all features.

We can make several observations from these results. First, the feature-specific models, used individually, improve over the baseline in almost all cases, although most improvements are not statistically significant, particularly for French to English. It is difficult to tell which of these models is best, due to the high variance across test set, smoothing procedure, and language pair. It is also hard to pick a clear winner between the 1-versus-all and multiple-feature smoothing methods: their average performance across all conditions is nearly identical.

The results from the clustered models are somewhat more stable. In all cases, scores increase at first as cluster size grows, then decrease. The optimum points are different for the two translation directions, but both are above the 1k threshold, which suggests that it is not worthwhile to train langauge models on portions of the corpus that are tagged with high-frequency document vectors.

Turning to the bottom blocks in each table, we can see that the combined models are generally better than the individual-feature models, indicating that the features capture complementary information. The linear combinations appear to be slightly better than the log-linear. This may be due to problems with MERT, since the log-linear combinations involve five language model features instead of just one as in all other approaches in tables 3 and 4.

Finally, the results for English to French translation appear to be somewhat better than for French to English translation, with the maximum gain over the baseline being approximately 0.5 BLEU points in the former case, and 0.3 BLEU points in the latter. The gains in both cases are statistically significant at the 0.95 level, however, according to paired bootstrap resampling (Koehn, 2004). This also holds for the results from the linear 1-versus-all combination in both translation settings; this technique offers a good combination of performance and efficiency, and is a good candidate for the best approach among the ones tested in this paper.

4 Related Work

As mentioned above, we are unaware of any previous work on translating structured documents in SMT. The closest related work is due to Finch and Sumita (2008). Like us, they use models that operate at sentence-level granularity, and are trained on corpus partitions. Unlike us, they modify all components of their log-linear model, they make only a single binary distinction (interrogative versus declarative sentences), and they use a maxent classifier to assign these properties to source sentences rather than relying on document structure.

Other relevant work is on domain adaptation. One way of viewing our approach is that it splits the source document into many micro-domains, and attempts to adapt to each. From this viewpoint, recent work on SMT adaptation (Foster and Kuhn, 2007; Koehn and Schroeder, 2007; Lü et al., 2007; Tam et al., 2007) is applicable, in addition to the IR approaches already mentioned. However, most of this work does not deal with adaptation along different axes simultaneously (for instance adapting to topic *and* genre). An exception is (Matsoukas et

al., 2009), who build a model for weighting phrase-pair joint counts during relative-frequency TM estimation that can depend on arbitrary features of the training corpus. Their feature weights are set discriminatively using a dev set and do not distinguish between sentences in that set, but it might be possible to extend the approach to allow for the sentence-level dependencies required for handling structural features.

5 Conclusion And Future Work

In this paper we have outlined a general approach for taking the structure of documents into account during statistical translation. This involves encoding structural information into feature vectors that characterize each source sentence, and building statistical models that are conditioned on this information.

Within this general framework, we propose two methods for modifying the language model, given a parallel training corpus that is also tagged with document-structure features. The first method concentrates on one feature at a time, partitioning the training corpus according to the different values it can take on and training an LM on each partition. Source sentences from the current test document are essentially translated using the LM that matches their feature value. To avoid sparsity problems on small corpus partitions, this is mixed with a global LM, with mixing weights set to optimum values determined from the current *source* text. A variant smoother mixes over LMs from all values the feature can take on. The smoothed feature-specific LMs obtained from this procedure are combined either linearly or log-linearly to produce the final structure-conditioned LM.

A second method basically treats feature vectors as atomic units and seeks to identify enough relevant text to be able to train a reliable LM for each. This is accomplished through a simple clustering algorithm that repeatedly pools the text corresponding to low-frequency vectors, using a minumum unigram likelihood drop criterion, until sufficient text is deemed to have been accumulated for each (at the cost of no longer distinguishing among the pooled vectors). Source sentences are then mapped to an appropriate smoothed LM, in a procedure similar to that used in the first method.

We tested both these methods on a corpus of structured Hansard documents, using five features capturing mostly complementary information. Results were mildly positive across the spectrum of approaches tested. The best, statistically significant, improvements were obtained from the first method outlined above, using a linear combination of feature-specific models.

Future Work

There are many possibilities for extending this exploratory work. The most obvious is to condition other models, for instance the TM, on document context. The methods outlined above could probably be adapted to this relatively easily, provided suitable smoothing techniques were used.

A harder challenge is to find better ways of conditioning on the document structure information. One interesting possibility would be to assume a latent hierarchical structure for the features in order to apply recent hierachical adaptation techniques (Finkel and Manning, 2009), suitably modified for multinomial language and translation models, possibly using MAP combinations (Bacchiani et al., 2004).

A related idea is to treat the features themselves as hints rather than performing the kind of hard matching used by the methods above. The multiple-feature smoothing methods are a step in this direction, and it would be interesting to apply a similar approach to the whole training corpus, in order to more accurately learn relations between features.

Finally, it would be interesting to experiment with other structured domains to determine if the Hansard is an outlier either in its availability of structured information or in the degree to which this is useful for translation.

References

Michel Bacchiani, Brian Roark, and Murat Saraclar. 2004. Language model adaptation with MAP estimation and the perceptron algorithm. In NAACL04 (NAA, 2004).

J. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized parallel backoff. In NAACL03 (NAA, 2003).

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In NAACL09 (NAA, 2009).

Andrew Finch and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, Columbus, June. WMT.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical Bayesian domain adaptation. In NAACL09 (NAA, 2009).

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In WMT07 (WMT, 2007).

Joshua Goodman. 2001. A bit of progress in language modeling. *Computer Speech and Language*.

Sasa Hasan and Hermann Ney. 2005. Clustered language models based on regular expressions for SMT. In *Proceedings of the 10th EAMT Conference*, Budapest, May.

Xiaodong He. 2007. Using word-dependent transition models in HMM based word alignment for statistical machine translation. In WMT07 (WMT, 2007).

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of the 10th EAMT Conference*, Budapest, May.

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

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In NAACL03 (NAA, 2003), pages 127–133.

P. Koehn. 2004. Statistical signicance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic.

Daniel Marcu, Lynn Carlson, and Maki Watanabe. 2000. The automatic translation of discourse structures. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP)*, Seattle, Washington, May.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

NAACL. 2003. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Edmonton, May.

NAACL. 2004. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Boston, May.

NAACL. 2009. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Boulder, June.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, July.

Yik-Cheung Tam, Ian Lane, and Tanja Schultz. 2007. Bilingual-LSA Based LM Adaptation for Spoken Language Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, June.

WMT. 2007. *Proceedings of the ACL Workshop on Statistical Machine Translation*, Prague, June.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In NAACL04 (NAA, 2004).

Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the International Conference on Computational Linguistics (COLING) 2004*, Geneva, August.