Dynamic Model Interpolation for Statistical Machine Translation

Andrew FINCH
NICT[†]-ATR[‡]
Kyoto, Japan
andrew.finch@atr.jp

Eiichiro SUMITA
NICT[†]-ATR[‡]
Kyoto, Japan
eiichiro.sumita@atr.jp

Abstract

This paper presents a technique for class-dependent decoding for statistical machine translation (SMT). The approach differs from previous methods of class-dependent translation in that the class-dependent forms of all models are integrated directly into the decoding process. We employ probabilistic mixture weights between models that can change dynamically on a segment-by-segment basis depending on the characteristics of the source segment. The effectiveness of this approach is demonstrated by evaluating its performance on travel conversation data. We used the approach to tackle the translation of questions and declarative sentences using class-dependent models. To achieve this, our system integrated two sets of models specifically built to deal with sentences that fall into one of two classes of dialog sentence: *questions* and *declarations*, with a third set of models built to handle the general class. The technique was thoroughly evaluated on data from 17 language pairs using 6 machine translation evaluation metrics. We found the results were corpus-dependent, but in most cases our system was able to improve translation performance, and for some languages the improvements were substantial.

1 Introduction

Topic-dependent modeling has proven to be an effective way to improve quality the quality of models in speech recognition (Iyer and Osendorf, 1994; Carter, 1994). Recently, experiments in the field of machine translation (Hasan and Ney, 2005; Yamamoto and Sumita, 2007; Finch et al. 2007, Foster and Kuhn, 2007) have shown that class-specific models are also useful for translation.

In the method proposed by Yamamoto and Sumita (2007), topic dependency was implemented by partitioning the data into sets before the decoding process commenced, and subsequently decoding these sets independently using different models that were specific to the class predicted for the source sentence by a classifier that was run over the source sentences in a pre-processing pass. Our approach is in many ways a generalization of this work. Our technique allows the use of multiple-model sets within the decoding process itself. The contributions of each model set can be controlled dynamically during the decoding through a set of interpolation weights. These weights can be changed on a sentence-by-sentence basis. The previous approach is, in essence, the case where the interpolation weights are either 1 (indicating that the source sentence is the same topic as the model) or 0 (the source sentence is a different topic). One advantage of our proposed technique is that it is a soft approach. That is, the source sentence can belong to multiple classes to varying degrees. In this respect our approach is similar to that of Foster and Kuhn (2007), however we used a probabilistic classifier to determine a vector of probabilities representing class-membership, rather than distance-based weights. These probabilities were used directly as the mixture weights for the respective models in an interpolated model-set. A second difference between our approach and that of Foster and Kuhn, is that we include a general model built from all of the data along with the set of class-specific models.

Our approach differs from all previous approaches in the models that are class-dependent. Hasan and Ney (2005) used only a class-dependent language model. Both Yamamoto and Sumita (2007) and Foster and Kuhn (2007), extended this to include the translation model. In our approach we combine all of the models, including the distortion and target length models, in the SMT system within a single framework.

The contribution of this paper is two-fold. The first is the proposal of a technique for combining

† National Institute for Science and Technology
‡ Advanced Telecommunications Research Laboratories

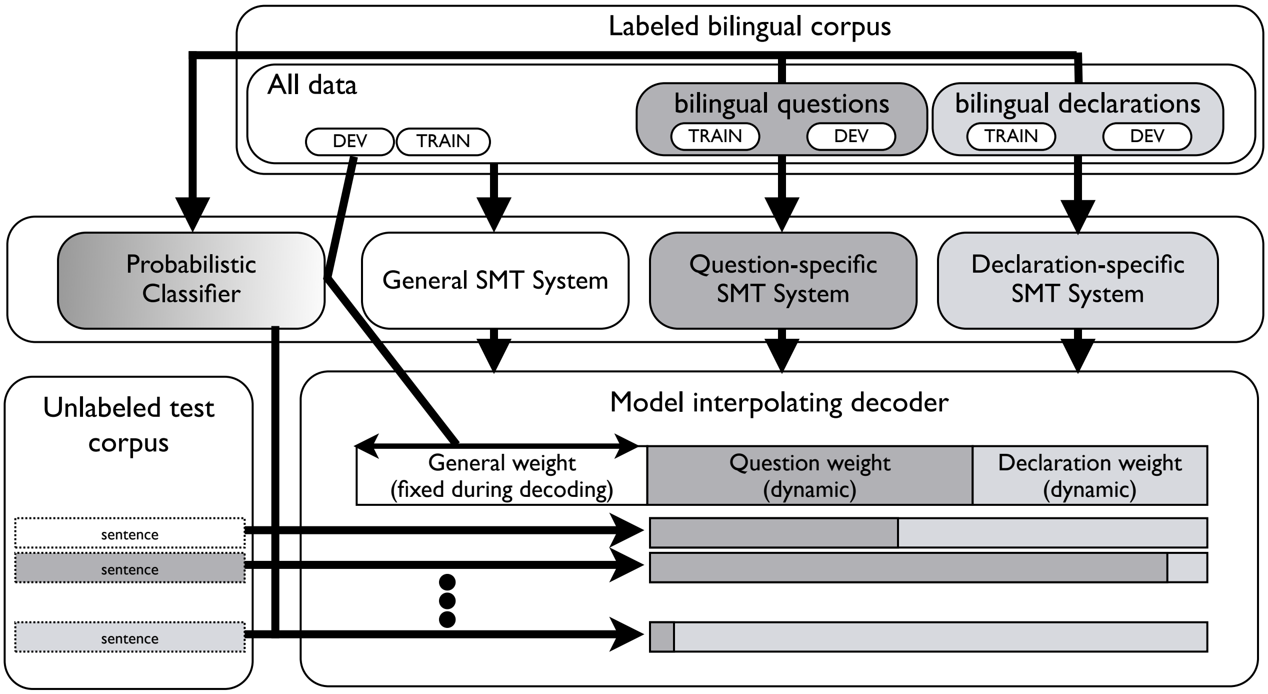

Figure 1. The architecture of the class-based SMT system used in our experiments

multiple SMT systems in a weighted manner to allow probabilistic soft weighting between topic-dependent models for all models in the system. The second is the application of this technique to improve the quality of dialog systems by building and combing class-based models for interrogative and declarative sentences.

For the purposes of this paper, we wish to make the distinction between interrogative sentences and those which are not. For the sake of simplicity of expression we will call those sentences which are interrogative, *questions* and those which are not, *declarations* for the remainder of this article.

The techniques proposed here were evaluated on a variety of different languages. We enumerate them below as a key: Arabic (ar), Danish (da), German (de), English (en), Spanish (es), French (fr), Indonesian (Malay) (id), Italian (it), Japanese (ja), Korean (ko), Malaysian (Malay) (ms), Dutch (nl), Portugese (pt), Russian (ru), Thai (th), Vietnamese (vi) and Chinese (zh).

2 System Overview

2.1 Experimental Data

To evaluate the proposed technique, we conducted experiments on a travel conversation corpus. The experimental corpus was the travel arrangement task of the BTEC corpus (Kikui et al., 2003) and used English as the target and each of the other languages as source languages. The training, development, and evaluation corpus statistics are shown in Table 1. The evaluation corpus had sixteen reference translations per sentence. This training corpus was also used in the IWSLT06 Evaluation Campaign on Spoken Language Translation (Paul 2006) J-E open track, and the evaluation corpus was used as the IWSLT05 evaluation set.

2.2 System Architecture

Figure 1 shows the overall structure of our system. We used punctuation (a sentence-final '?' character) on the target-side as the ground truth as to the class of the target sentence. Neither punctuation nor case information was used for any other purpose in the experiments. The data were partitioned into classes, and further sub-divided into training and development sets for each class. 1000 sentences were set aside as development data, and the remainder was used for training. Three complete SMT systems were built: one for each class, and one on the data from both classes. A probabilistic classifier (described in the next section) was also trained from the full set of training data.

The machine translation decoder used is able to linearly interpolate all of the models models from

	Questions + Decls.		Questions		Declarations		Test
	Train	Dev	Train	Dev	Train	Dev	
Sentences	161317	1000	69684	1000	90633	1000	510
Words	1001671	6112	445676	6547	549375	6185	3169

Table 1. The corpus statistics of the target language corpus (en). The number of sentences is the same as these values for all source languaes. The number of words in the source language differs, and depends on the segmentation granularity.

all of the sub-systems according to a vector of interpolation weights supplied for each source word sequence to be decoded. To do this, prior to the search, the decoder must first merge the phrase-tables from each sub-system. Every phrase from all of the phrase-tables is used during the decoding. Phrases that occur in one sub-system's table, but do not occur in another sub-system's table will be used, but will receive no support (zero probability) from those sub-systems that did not acquire this phrase during training. The search process proceeds as in a typical multi-stack phrase-based decoder. The weight for the general model was set by tuning the parameter on the general development set in order to maximize performance in terms of BLEU score. This weight determines the amount of probability mass to be assigned to the general model, and it remains fixed during the decoding of all sentences. The remainder of the probability mass is divided among the class-specific models dynamically sentence-by-sentence at run-time. The proportion that is assigned to each class is simply the class membership probability of the source sequence assigned by the classifier.

3 Question Prediction

3.1 Outline of the Problem

Given a source sentence of a particular class (interrogative or declarative in our case), we wish to ensure that the target sentence generated is of an appropriate class. Note that this does not necessarily mean that given a question in the source, a question should be generated in the target. However, it seems reasonable to assume that, intuitively at least, one should be able to generate a target question from a source question, and a target declaration from a source declaration. This is reasonable because the role of a machine translation engine is not to be able to generate every possible translation from the source, but to be able to generate one acceptable translation. This assumption leads us to two plausible ways to proceed.

1. To predict the class of the source sentence, and use this to constrain the decoding process used to generate the target
2. To predict the class of the target

In our experiments, we chose the second method, as it seemed the most correct, but feel there is some merit in both strategies.

3.2 The Maximum Entropy Classifier

We used a Maximum Entropy (ME) classifier to determine which class to which the input source sentence belongs using a set of lexical features. That is, we use the classifier to set the mixture weights of the class-specific models. In recent years such classifiers have produced powerful models utilizing large numbers of lexical features in a variety of natural language processing tasks, for example Rosenfeld (1996). An ME model is an exponential model with the following form:

$$p(t,c) = \gamma \prod_{k=0}^{K} \alpha_k^{f_k(c,t)} p_0$$

where:

t	is the class being predicted;
c	is the context of t;
γ	is a normalization coefficient;
K	is the number of features in the model;
α_k	is the weight of feature f_k;
f_k	are binary feature functions;
p_0	is the default model

<s> where　　　　　　is the
<s> where is
<s> where is the　　　is the station </s>
is　　　　　　　the station </s>
the　　　　　　　station </s>

Figure 2. The set of n-gram ($n \leq 3$) features extracted from the sentence <s> where is the station </s> for use as predicates in the ME model to predict target sentence class.

We used the set of all n-grams ($n \leq 3$) occurring in the source sentences as features to predict the sentence's class. Additionally we introduced beginning of sentence tokens (<s>) and end of sentence tokens into the word sequence to distinguish n-grams occurring at the start and end of sentences from those occurring within the sentence. This was based on the observation that "question words" or words that indicate that the sentence is a question will frequently be found either at the start of the sentence (as in the wh- <what, where, when> words in English or the -kah words in Malay <apakah, dimanakah, kapankah>), or at the end of the sentence (for example the Japanese "ka" or the Chinese "ma"). In fact, in earlier models we used features consisting of n-grams occurring only at the start and end of the source sentence. These classifiers performed quite well (approximately 4% lower than the classifiers that used features from all of the n-grams in the source), but an error analysis showed that n-grams from the interior of the sentence were necessary to handle sentences such as "excuse me please where is ...". A simple example sentence and the set of features generated from the sentence is shown in Figure 2.

We used the ME modeling toolkit of (Zhang, 2004) to implement our ME models. The models were trained by using L-BFGS parameter estimation, and a Gaussian prior was used for smoothing during training.

3.3 Forcing the target to conform

Before adopting the mixture-based approach set out in this paper, we first pursued an obvious and intuitively appealing way of using this classifier. We applied it as a filter to the output of the decoder, to force source sentences that the classifier predicts should generate questions in the target to actually generate questions in the target. This approach was unsuccessful due to a number of issues.

Source Language	English Punctuation	Own Punctuation
ar	98.0	N/A
da	97.3	98.0
de	98.1	98.6
en	98.9	98.9
es	96.3	96.7
fr	97.7	98.7
id	97.9	98.5
it	94.9	95.4
ja	94.1	N/A
ko	94.2	99.4
ms	98.1	99.0
nl	98.1	99.0
pt	96.2	96.0
ru	95.9	96.6
th	98.2	N/A
vi	97.7	98.0
zh	93.2	98.8

Table 2. The classifcation accuracy (%) of the classifier used to predict whether or not an input sentence either is or should give rise to a question in the target.

We took the n-best output from the decoder and selected the highest translation hypothesis on the list that had agreement on class according to source and target classifiers. The issues we encountered included, too much similarity in the n-best hypotheses, errors of the MT system were correlated with errors of the classifier, and the number of cases that were corrected by the system was small <2%. As a consequence, the method proposed in this paper was preferred.

4 Experiments

4.1 Experimental Conditions

Decoder

The decoder used to in the experiments, CleopATRa is an in-house phrase-based statistical decoder that can operate on the same principles as the PHARAOH (Koehn, 2004) and MOSES (Koehn et

Source	BLEU	NIST	WER	PER	GTM	METEOR
ar	0.4457 (0.00)	8.9386 (0.00)	0.4458 (0.00)	0.3742 (0.00)	0.7469 (0.00)	0.6766 (0.00)
da	0.6640 (0.64)	11.4500 (1.64)	0.2560 (0.08)	0.2174 (2.42)	0.8338 (0.68)	0.8154 (1.23)
de	0.6642 (0.79)	11.4107 (0.44)	0.2606 (2.18)	0.2105 (0.14)	0.8348 (-0.13)	0.8132 (-0.07)
es	0.7345 (0.00)	12.1384 (0.00)	0.2117 (0.00)	0.1668 (0.00)	0.8519 (0.00)	0.8541 (0.00)
fr	0.6666 (0.95)	11.7443 (0.63)	0.2548 (4.82)	0.2172 (6.50)	0.8408 (0.48)	0.8293 (1.29)
id	0.5295 (9.56)	10.3459 (4.11)	0.3899 (21.17)	0.3239 (4.65)	0.7960 (1.35)	0.7521 (2.35)
it	0.6702 (1.01)	11.5604 (0.41)	0.2590 (3.25)	0.2090 (0.62)	0.8351 (0.36)	0.8171 (0.05)
ja	0.5971 (3.47)	10.6346 (2.56)	0.3779 (5.53)	0.2842 (2.80)	0.8125 (0.74)	0.7669 (0.67)
ko	0.5898 (1.78)	10.2151 (1.31)	0.3891 (0.74)	0.3138 (-0.10)	0.7880 (0.36)	0.7397 (0.35)
ms	0.5102 (10.19)	9.9775 (2.75)	0.4058 (18.53)	0.3355 (3.59)	0.7815 (0.18)	0.7247 (2.49)
nl	0.6906 (2.55)	11.9092 (1.47)	0.2415 (3.21)	0.1872 (1.73)	0.8548 (0.39)	0.8399 (0.36)
pt	0.6623 (0.35)	11.6913 (0.26)	0.2549 (2.52)	0.2110 (2.68)	0.8396 (0.02)	0.8265 (-0.07)
ru	0.5877 (0.34)	10.1233 (-1.10)	0.3447 (1.99)	0.2928 (1.71)	0.7900 (0.15)	0.7537 (-0.40)
th	0.4857 (1.50)	9.5901 (1.17)	0.4883 (-0.23)	0.3579 (2.03)	0.7608 (0.45)	0.7104 (1.23)
vi	0.5118 (0.67)	9.8588 (1.85)	0.4274 (-0.05)	0.3301 (0.12)	0.7806 (1.05)	0.7254 (0.43)
zh	0.5742 (0.00)	10.1263 (0.00)	0.3937 (0.00)	0.3172 (0.00)	0.7936 (0.00)	0.7343 (0.00)

Table 3. Performance results translating from a number of source languages into English. Figures in parentheses are the percentage improvement in the score relative to the original score. Bold-bordered cells indicate those conditions where performance degraded. White cells indicate the proposed system's performance is significanly different from the baseline (using 2000-sample bootstrap resampling with a 95% confidence level). TER scores were not tested for significance due to technical difficulties. ar, es and zh were also omitted since the systems were identical.

al, 2007) decoders. The decoder was configured to produce near-identical output to MOSES for these experiments. The decoder was modified in order to handle multiple-sets of models, accept weighted input, and to incorporate the dynamic interpolation process during the decoding.

Practical Issues

Perhaps the largest concerns about the proposed approach come from the heavy resource requirements that could potentially occur when dealing with large numbers of models. However, one important characteristic of the decoder used in our experiments is its ability to leave its models on disk, loading only the parts of the models neces-

Source	Baseline	No Classifier	Hard	Proposed
ar	0.4457 (0.00)	0.4457 (0.00)	0.4457 (0.00)	0.4457
da	0.6598 (0.64)	0.6647 (-0.11)	0.6591 (0.74)	0.664
de	0.6590 (0.79)	0.6651 (-0.14)	0.6634 (0.12)	0.6642
es	0.7345 (0.00)	0.7345 (0.00)	0.7345 (0.00)	0.7345
fr	0.6603 (0.95)	0.6594 (1.09)	0.6605 (0.92)	0.6666
id	0.4833 (9.56)	0.5029 (5.29)	0.5276 (0.36)	0.5295
it	0.6635 (1.01)	0.6660 (0.63)	0.6644 (0.87)	0.6702
ja	0.5771 (3.47)	0.5796 (3.02)	0.5667 (5.36)	0.5971
ko	0.5795 (1.78)	0.5837 (1.05)	0.5922 (-0.41)	0.5898
ms	0.4630 (10.19)	0.5015 (1.73)	0.5057 (0.89)	0.5102
nl	0.6734 (2.55)	0.6902 (0.06)	0.6879 (0.39)	0.6906
pt	0.6600 (0.35)	0.6643 (-0.30)	0.6598 (0.38)	0.6623
ru	0.5857 (0.34)	0.5885 (-0.14)	0.5844 (0.56)	0.5877
th	0.4785 (1.50)	0.4815 (0.87)	0.4831 (0.54)	0.4857
vi	0.5084 (0.67)	0.5095 (0.45)	0.5041 (1.53)	0.5118
zh	0.5742 (0.00)	0.5742 (0.00)	0.5742 (0.00)	0.5742

Table 4. Performance results comaparing our proposes method with other techniques. The column labeled ‘Baseline’ is the same as in Table 3, for reference. The column lableled ‘No Classifier’, is the same system as our proposed method, except that the classifier was replaced with a default model that assigned a class membership probability of 0.5 in every case. The column labeled ‘Hard’ corresponds to a system that used hard weights (either 1 or 0) for the class-dependent models. The column labeled ‘Proposed’ are the results from our proposed method. Figures in parentheses represent the percentage improvement of the proposed method’s score relative to the alternative method. Cells with bold borders indicate those conditions where performance was degraded.

sary to decode the sentence in hand. This reduced the memory overhead considerably when loading multiple models, without noticeably affecting decoding time. Moreover, it is also possible to pre-compute the interpolated probabilities for most of the models for each sentence before the search commences, reducing both search memory and processing time.

Decoding Conditions

For tuning of the decoder's parameters, minimum error training (Och 2003) with respect to the BLEU score using was conducted using the respective development corpus. A 5-gram language model, built using the SRI language modeling toolkit (Stolcke, 1999) with Witten-Bell smoothing was used. The model included a length model, and also the simple distance-based distortion model used by the PHARAOH decoder (Koehn, 2004).

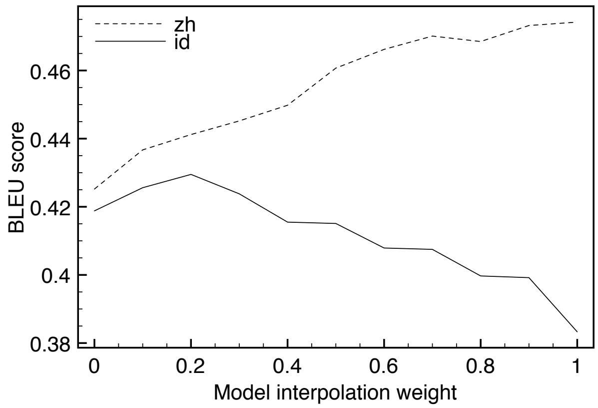

Figure 3. Graph showing the BLEU score on the developmment set plotted against the general model's interpolation weight (a weight of 0 meaning no contribution from the general model) for two systems in our experiments.

Tuning the interpolation weights

The interpolation weights were tuned by maximizing the BLEU score on the development set over a set of weights ranging from 0 to 1 in increments of 0.1. Figure 1 shows the behavior of two of our models with respect to their weight parameter.

Evaluation schemes

To obtain a balanced view of the merits of our proposed approach, in our experiments we used 6 evaluation techniques to evaluate our systems. These were: BLEU (Papineni, 2001), NIST (Doddington, 2002), WER (Word Error Rate), PER (Position-independent WER), GTM (General Text Matcher), and METEOR (Banerjee and Lavie, 2005).

4.2 Classification Accuracy

The performance of the classifier (from 10-fold cross-validation on the training set) is shown in Table 2. We give classification accuracy figures for predicting both source (same language) and target (English) punctuation. Unsurprisingly, all systems were better at predicting their own punctuation. The poorer scores in the table might reflect linguistic characteristics (perhaps questions in the source language are often expressed as statements in the target), or characteristics of the corpus itself. For all languages the accuracy of the classifier seemed satisfactory, especially considering the possibility of inconsistencies in the corpus itself (and therefore our test data for this experiment).

4.3 Translation Quality

The performance of the SMT systems are shown in Table 3. It is clear from the table that for most of the experimental conditions evaluated the system outperformed a baseline system that consisted of an SMT system trained on all of the data. For those metrics in which performance degraded, in all-but-one the results were statistically insignificant, and in all cases most of the other MT evaluation metrics showed an improvement. Some of the language pairs showed striking improvements, in particular both of the Malay languages id and ms improved by over 3.5 BLEU points each using our technique. Interestingly Dutch, a relative of Malay, also improved substantially. This evidence points to a linguistic explanation for the gains. Malay has very simple and regular question structure, the question words appear at the front of question sentences (in the same way as the target language) and do not take any other function in the language (unlike the English "do" for example). Perhaps this simplicity of expression allowed our class-specific models to model the data well in spite of the reduced data caused by dividing the data. Another factor might be the performance of the classifier which was high for all these languages (around 98%). Unfortunately, it is hard to know the reasons behind the variety of scores in the table. One large factor is likely to be differences in corpus quality, and also the relationship between the source and target corpus. Some corpora are direct translations of each other, whereas others are translated through another language. Chinese was one such language, and this may explain why we were unable to improve on the baseline for this language even though we were very successful for both Japanese and Thai, which are relatives of Chinese.

4.4 Comparison to Previous Methods

We ran an experiment to compare our proposed method to an instance of our system that used hard weights. The aim was to come as close as possible within our framework to the system proposed by Yamamoto and Sumita (2007). We used weights of 1 and 0, instead of the classification probabilities to weight the class-specific models. To achieve this, we thresholded the probabilities from the classifier such that probabilities >0.5 gave a weight of 1, otherwise a weight of 0 was used. The performance of this system is shown in Table 4 under the column heading 'Hard'. In all-but-one of the conditions this system was outperformed by or equal to the proposed approach.

The column labeled "No Classifier" in Table 4 illustrates the effectiveness of the classifier in our system. These results show the effect of using equal weights (0.5) to interpolate between the Question and Declaration models. This system, although not as effective as the system with the classifier, gave a respectable performance.

5 Conclusion

In this paper we have presented a technique for combining all models from multiple SMT engines into a single decoding process. This technique allows for topic-dependent decoding with probabilistic soft weighting between the component models. We demonstrated the effectiveness of our approach on conversational data by building class-specific models for interrogative and declarative sentence classes. We carried out an extensive evaluation of the technique using a large number of language pairs and MT evaluation metrics. In most cases we were able to show significant improvements over a system without model interpolation, and for some language pairs the approach excelled. The best improvement of all the language pairs was for Malaysian (Malay)-English which outperformed the baseline system by 4.7 BLEU points (from 0.463 to 0.510). In future research we would like to try the approach with larger sets of models, and also (possibly overlapping) subsets of the data produced using automatic clustering methods.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. *ACL-2005: Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65-72.

David Carter, 1994. Improving Language Models by Clustering Training Sentences, *Proc. ACL*, pp. 59-64.

J. Civera and A. Juan. Domain adaptation in statistical machine translation with mixture modelling. *In Proceedings of the Second Workshop on ACL Statistical Machine Translation*, pp. 177-180, Prague,

Czech Republic, June 2007.Andrew Finch, Etienne Denoual, Hideo Okuma, Michael Paul, Hirofumi Yamamoto, Keiji Yasuda, Ruiqiang Zhang, and Eiichiro Sumita. 2007. The NICT/ATR speech translation system for IWSLT 2007. *IWSLT 2007*, Trento, Italy.

George Doddington. 2002 Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proc. of Human Language Technology Conference*, San Diego, California, pp. 138-145.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. *In Proceedings of the Second Workshop on Statistical Machine Translation*, ACL, pp. 128-135, Prague, Czech Republic,

Sasa Hasan and Hermann Ney. 2005. Clustered Language Models Based on Regular Expressions for SMT, *Proc. EAMT*, Budapest, Hungary.

Rukmini Iyer and Mari Ostendorf. 1994. Modeling Long Distance Dependence in Language: Topic mixture versus dynamic cache models, *IEEE Transactions on Speech and Audio Processing*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. *In Proceedings of the Human Language Technology Conference 2003*, Edmonton, Canada.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. Machine translation: from real users to research: *6th conference of AMTA*, Washington, DC, Springer Verlag, pp. 115-124.

Philipp Koehn, 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, Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation, *ACL 2007: proceedings of demo and poster sessions*, Prague, Czech Republic, pp. 177-180.

Franz J. Och, Hermann Ney, 2003. A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, No. 1, Vol. 29, pp. 19-51.

Franz J. Och, 2003. Minimum error rate training for statistical machine trainslation, *Proc. ACL*.

Michael Paul, 2006. Overview of the IWSLT 2006 Evaluation Campaign, *IWSLT 2006*.

Kishore Papineni, Salim Roukos, Todd Ward, & Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. *IBM Research Report, RC22176*, September 17.

Ronald Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation, *Proceedings of Association for Machine Translation in the Americas*.

Andreas Stolcke. 1999. SRILM - An Extensible Language Model Toolkit. http://www.speech.sri.com/projects/srilm/

Hirofumi Yamamoto and Eiichiro Sumita. 2007. Bilingual cluster based models for statistical machine translation. *EMNLP-CoNLL-2007*, Prague, Czech Republic; pp. 514-523.

Le Zhang. 2004. Maximum Entropy Modeling Toolkit for Python and C++, [On-line].