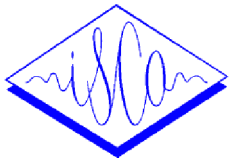
*Proceedings of IWSLT 2009, Tokyo - Japan*





# The University of Washington Machine Translation System for IWSLT 2009

*Mei Yang, Amittai Axelrod, Kevin Duh, Katrin Kirchhoff*

Department of Electrical Engineering
University of Washington, Seattle
{yangmei,amittai,duh,katrin}@ee.washington.edu

## Abstract

This paper describes the University of Washington's system for the 2009 International Workshop on Spoken Language Translation (IWSLT) evaluation campaign. Two systems were developed, one each for the BTEC Chinese-to-English and Arabic-to-English tracks. We describe experiments with different preprocessing and alignment combination schemes. Our main focus this year was on exploring a novel semi-supervised approach to N-best list reranking; however, this method yielded inconclusive results.

## 1. Introduction

The University of Washington submitted systems for two translation tasks in the 2009 IWSLT shared evaluation campaign: the BTEC Chinese-to-English and Arabic-to-English tracks. Our main interest this year was in testing a novel method for semi-supervised reranking of N-best lists, which has previously shown improvements on 2007 IWSLT data. We additionally explored different preprocessing schemes for both language pairs, as well as methods for combining phrase tables based on different word alignments. In the following sections we first describe the data, general baseline system and post-processing steps, before describing language-pair specific methods and the semi-supervised reranking method.

## 2. Corpora and Preprocessing

As mandated by the evaluation guidelines, the only data that was used for system development was the official data provided by IWSLT. Training data for the BTEC tasks consisted of approximately 20,000 sentence pairs in both the Chinese-English and Arabic-English tracks. We used the combined development datasets (about 500 sentences each) for initial system tuning, except for the IWSLT 2008 eval set, which we used as a held-out set for testing generalization performance.

We performed initial corpus preprocessing with the provided scripts, i.e. the English half of each parallel corpus was processed by lowercasing and tokenizing all punctuation and possessive clitics. Although the Chinese data came in segmented form, we also tested alternative segmentation methods. For Arabic, we compared various tokenization schemes. These variants are further described in the system-specific sections below.

The training corpus was additionally processed by filtering sentence pairs according to the ratio of the source and target sentence lengths, in order to eliminate mismatched sentence pairs that would skew the trained models. A 9:1 ratio was used; however, this did not eliminate any sentences from the Chinese-English corpus and only 52 sentences from the Arabic-English corpus.

## 3. Basic System Overview

### 3.1. Translation Model

Our baseline system for this year's task is a state-of-the-art, two-pass phrase-based statistical machine translation system, based on a log-linear translation model [7].

$$e^* = argmax_e p(e|f) = argmax_e \{\sum_{k=1}^{K} \lambda_k \phi_k(e,f)\} \quad (1)$$

where $e$ is an English sentence, $f$ is a foreign sentence, $\phi_k(e,f)$ is a feature function defined on both sentences, and $\lambda_k$ is a feature weight. We trained this model within the Moses development and decoding framework [8]. The feature functions used in this year's system include:

- two phrase-based translation scores, one for each translation direction
- two lexical translation scores, one for each translation direction
- six lexical reordering scores
- word count penalty
- phrase count penalty
- distortion penalty
- language model score

For a segmentation of source and target sentences into phrases, $f = \bar{f}_1, \bar{f}_2, ..., \bar{f}_M$ and $e = \bar{e}_1, \bar{e}_2, ..., \bar{e}_M$, the phrasal translation score for $\bar{e}$ given $\bar{f}$ is computed as

$$P(\bar{e}|\bar{f}) = \frac{count(\bar{e}, \bar{f})}{count(\bar{f})} \quad (2)$$

i.e. as the relative frequency estimate from the phrase-segmented training corpus. The lexical score is computed as

$$Score_{lex}(\bar{e}|\bar{f}) = \prod_{j=1}^{J} \frac{1}{|\{j|a(i)=j\}|} \sum_{a(i)=j}^{I} p(f_j|e_i) \quad (3)$$

where $j$ ranges over words in phrase $\bar{f}$ and $i$ ranges over words in phrase $\bar{e}$. The lexical reordering model estimates the probablity of a sequence of orientations $o = (o_1, o_2, \ldots, o_M)$

$$P(o|f,e) = \prod_{i=1}^{M} P(o_i|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}) \quad (4)$$

where each $o_i$ takes one of the three values: monotone, swap, and discontinuous. This model adds six feature functions to the overall log-linear model: for each of the three orientations, the orders of the source phrase with respect to both the previous and the next source phrase are considered. The feature scores are again estimated by relative frequency.

The training corpus was word-aligned by GIZA++; subsequently, phrases were extracted using the technique in [6] and as implemented in the Moses training scripts [8]. We also used an alternative word-alignment based on the MTTK [6] implementation of an HMM-based word-to-phrase alignment model with bigram probabilities. This yielded mixed results, as described in later sections.

Word count and phrase count penalties are constant weights added for each word/phrase used in the translation; the distortion penalty is a weight that increases in proportion to the number of positions by which phrases are reordered during translation. The language models used are n-gram models as further described below. The weights for these scores were optimized using an in-house implementation of the minimum-error rate training (MERT) procedure developed in [9]. Our optimization criterion was the BLEU score on the available development set.

### 3.2. Language Models

For first-pass decoding we used trigram language models. We built all of our language models using the SRILM toolkit [4] with modified Kneser-Ney discounting and interpolating all n-gram estimates of order $> 1$. Due to the small size of the training corpus, we experimented with lowering the minimum count requirement to 1 for all n-grams. This yielded different results for the two different tasks, which are further described below.

### 3.3. Decoding

Our system used the Moses decoder to generate 100-best distinct output hypotheses per input sentence during the first translation pass. For the second pass, the N-best lists were rescored with additional models: higher-order language models, POS-based language models, and sentence-type specific POS language models. These yielded mixed results depending on the language pair and are described in the system-specific sections.

### 3.4. Postprocessing

As a first postprocessing step, all untranslated source language words are deleted. Our two-pass machine translation system produces lowercase English output with tokenized punctuation and possessives. In order to match the evaluation guidelines, we post-processed the output by re-attaching the possessive particle and restoring true case. Truecasing is done by a noisy-channel model as implemented in the *disambig* tool in the SRILM package. It uses a 4-gram model trained over a mixed-case representation of the BTEC training corpus and a probabilistic mapping table for lowercase-uppercase word variants. The first letter at the beginning of each sentence was uppercased deterministically.

## 4. Chinese → English

### 4.1. Preprocessing

Although the Chinese training data was pre-segmented we nonetheless explored other segmentation tools. First, we used the Stanford segmenter [2] to resegment the Chinese data, as it provides templates for annotating numbers and dates, potentially aiding in word alignment and phrase extraction. In another experiment, an in-house tool [11] was used to simply markup dates and numbers in the pre-segmented BTEC data. Third, we developed our own markup tool for numbers. In both Chinese and English, numbers are represented by a combination of a limited set of number words. A simple method for detecting numbers is to first obtain a set of number words and then search for subsentential chunks that are comprised of only number words. These chunks are considered a single number and are replaced by a special tag. This might prevent number translation errors due to wrong word segmentations. We first replace all numbers in a sentence by such tags, translate the sentence, and restore the number tag based on a look-up table. However, we did not notice a significant improvement in translation quality. Finally, we investigated a simple character-based segmentation approach in which each Chinese character was treated as a single word. This greatly reduces the size of the Chinese vocabulary but increases the difficulties of word alignment training because of the increased ambiguity for aligning each Chinese word. We tested the segmentation on the development set but found it led to worse performance compared to the original word segmentation. As a final experiment on preprocessing we also trained a system on data which had been stripped of all punctuation marks, mimicking the `no_case+no_punc` track, but this did not improve our baseline system either.

### 4.2. Word Alignment and Phrase Tables

As mentioned above, we used two different methods for word alignment. In addition to the standard GIZA++ training procedure we used a word-to-phrase HMM-based alignment model with bigram probabilities, using the MTTK package. Subsequently, alignments trained in each direction were combined using the grow-diag-final heuristic described in [6]. Taken by itself, the MTTK-based alignment resulted in a system that performed worse than the GIZA++-based system, resulting in a 2-point drop in BLEU on the held-out set. However, we experimented with combining both tables. To this end the individual tables were combined into a single table containing the 11 standard features (phrasal, lexical, and reordering scores and phrase penalty) plus two additional binary features that indicate which alignment model produced each entry in the phrase table. The weights for these features were optimized along with all other features in the first-pass MERT tuning.

### 4.3. Language Models

We decreased the minimum required count for n-grams with order $> 1$ to 1. This led to larger n-gram coverage and an increase in BLEU of 1 point on the held-out set.

### 4.4. Rescoring

For second-pass rescoring we evaluated higher-order n-gram models (4-grams and 5-grams) a part-of-speech (POS) based language model and sentence-type specific POS models. As a POS model, we tested both 4-gram and 5-gram language models trained on a POS annotation of the training data and n-best lists using Ratnaparkhi's maximum-entropy tagger [10]. None of these improved the performance. Questions and statements usually have quite different syntactic structures in both Chinese and English. In order to capture these differences, sentence-type specific POS models were trained, one for questions, one for statements. These were used in the second pass to rerank questions and non-questions, respectively. The sentence type was determined from the punctuation on the source side. This led to a small improvement in performance

### 4.5. Final Systems

For the final primary system, the development data was added to the training data, however, weights were not retuned. Our contrastive submission used a novel re-ranking method, detailed in Section 6. The training corpus, pre- and post-processing methods, and first-pass system were the same as in the primary system.

## 5. Arabic → English

### 5.1. Preprocessing

We preprocessed the Arabic data by using the the Columbia University MADA and TOKAN tools [5]. We compared two tokenization schemes: the first splits off the conjunctions *w+*, *f+*, the particles *l+*, the *b+* preposition and the definite article *Al+*. It also normalizes different variants of alif, final yaa and taa marbuta. The second scheme (equivalent to TOKAN's D2 scheme) does not split off *Al+* but instead separates the prefix *s+*. Differences between the two schemes were slight; the first scheme yielded a 0.2 increase in BLEU on the held-out set.

### 5.2. Word Alignment and Phrase Tables

As in the Chinese-English system, we trained word alignments using both GIZA++ and MTTK. We found that MTTK gave significantly worse results (by 6 BLEU points) compared to GIZA++, so it was not used either in isolation or for phrase table combination.

### 5.3. Language Models

As mentioned in the system overview, we tried lowering the minimum count threshold for the 3- and 4-grams in the language model (from 2 to 1). This greatly increased the language model's coverage from 17k to 95k 3-grams, and from 15k to 120k 4-grams. However, the overall system score decreased by 0.5 BLEU points and so our Arabic-to-English submission uses the default SRILM cutoffs. Note that this is in contrast with our Chinese-to-English results.

### 5.4. Rescoring

For rescoring we investigated higher-order n-gram models, POS-based 4-gram and 5-gram language models as well as sentence-type specific POS models. In this case, however, the baseline performance was not improved by either type of model.

## 6. Semi-Supervised Reranking

For our contrastive Chinese-to-English system, we enriched the baseline system with a semi-supervised re-ranker that utilizes information inherent in the test set's N-best lists. The goal is to "adapt" a general re-ranker to each test list independently and to rescore it using the adapted re-ranker. Such "local learning" approaches may outperform a global re-ranker that was trained to optimize performance globally on an entire development set.

Our semi-supervised reranking approach is based on a modification of the RankBoost [3] learning algorithm, a state-of-the-art machine learning technique for reranking. RankBoost treats the re-ranking problem as a problem of binary classification on pairs of hypotheses (hypothesis $x$ is ranked higher than $y$ or vice versa) and maintains a weight

distribution over labeled instances in the training set. It then iteratively trains a weak ranker on the labeled instances and adjusts the weight distribution according to the correctness of the classification decisions made by the weak ranker, decreasing the weights for correctly classified samples, and increasing the weights for wrongly classified samples. Finally, individual rankers are combined in a weighted fashion. Letting our ranking function be $f()$, where hypotheses with high $f$ values are ranked higher, RankBoost minimizes the following objective:

$$\sum_{<x,y>\in P_L} exp(-(f(x) - f(y))) \quad (5)$$

The set $P_L$ (the labeled set) consists of all pairs of elements in the lists where $x$ is ranked higher than $y$ (according to the true labels). The difference $f(x) - f(y)$ can also be thought of as a margin, which the learner aims to maximize. In order to adapt such a ranker to a given test N-best list, we add a second criterion that computes a "pseudo-margin" from pairs of hypotheses in the test list:

$$\sum_{(x,y)\in P_L} exp(-(f(x) - f(y))) \quad (6)$$

$$+ \quad \beta * \sum_{<i,j>\in P_U} exp(|f(i) - f(j)|) \quad (7)$$

Here, the set $P_U$ (the unlabeled set) consists of all pairs of hypotheses from the test list in focus; $\beta$ is a trade-off parameter balancing the contributions of the labeled vs. unlabeled data. This ranking objective corresponds to the cluster assumption used in semi-supervised classification [1]. The effect is to force the ranker to be more confident in teasing apart (clustering) good vs. bad hypotheses in the test list.

In our context of reranking N-best lists produced by a machine translation system, the labeled and unlabeled sets are determined as follows: The set $P_L$ is produced by first computing the smoothed sentence-level BLEU score for each hypothesis in a given N-best list. If sentence-level BLEU score of hypothesis $x$ is greater than that of hypothesis $y$ by a threshold $\tau$, the pair of hypotheses receives a positive label, else it receives a negative label. If the BLEU difference is less than $\tau$, the hypotheses are considered tied and are not used for training. The complete $P_L$ set consists of all valid hypothesis pairs extracted from the development set. The set $P_U$ is made up of all valid hypothesis pairs (determined according to the same threshold $\tau$) from a single test N-best list. Thus, a different semi-supervised ranker is trained for every N-best list in the test set.

The weak rankers are rankers based on individual feature function scores in the log-linear model. The final ranking function is

$$f(x) = \sum_i \theta_i h_i(x) \quad (8)$$

where $i$ ranges over all iterations performed and $\theta_i$ is a weight proportional to the ranker's accuracy.

| System | baseline | +reranking |
|---|---|---|
| BLEU | 0.368 | 0.362 |
| NIST | 6.596 | 5.890 |
| PER | 0.444 | 0.432 |
| TER | 0.434 | 0.400 |
| WER | 0.521 | 0.498 |
| METEOR | 0.636 | 0.638 |
| GTM | 0.672 | 0.658 |
| F1 | 0.681 | 0.696 |
| PREC | 0.699 | 0.738 |
| RECL | 0.664 | 0.658 |

Table 1: System performance for the Chinese-English baseline system vs. semi-supervised reranking, truecased version.

In the past we have seen substantial improvements from this method over both standard RankBoost and MERT on the IWSLT 2007 Italian-English and Arabic-English data (improvements in BLEU of 3.1 and 1.8 points, respectively, for baselines of 21.2 and 24.3). This motivated us to apply this method to this year's task as well. The $\beta$ parameter was set to 1, giving equal weight to labeled and unlabeled data. The $\tau$ parameter was optimized on the IWSLT 2008 eval set and was set to 55. On the eval08 set (our development set), the BLEU score increased slightly from 41.9 to 42.4. The complete set of scores for the 2009 eval set is shown in Table 1. We observe that n-gram based evaluation scores like BLEU and NIST decrease slightly whereas PER, TER, WER and Precision improve.

Our final primary system is trained on all the IWSLT09 data, including the development data. Therefore, the reranking algorithm cannot be retrained for this system since no development set is available for training the boosted classifier. If we use the classifier trained for the system shown in Table 1 and apply it to our final primary system as is, we obtain the results shown in Table 2. We see that the BLEU score drops slightly while PER still shows a slight improvement. Overall it seems that the reranker increases precision at the expense of recall.

## 7. Results

The official evaluation results for our primary systems are shown in Tables 3 and 4. Not that the primary system for Chinese-English was obtained after a final training pass where all development data had been added to the training data. It is obvious that the largest single gain derives from using all available data for training rather than from better reranking methods.

## 8. Conclusions

We have presented our systems for the IWSLT09 Arabic-English and Chinese-English BTEC tasks. In contrast to

| System | baseline | +reranking |
|---|---|---|
| BLEU | 0.406 | 0.401 |
| NIST | 7.048 | 6.368 |
| PER | 0.424 | 0.417 |
| TER | 0.424 | 0.388 |
| WER | 0.500 | 0.481 |
| METEOR | 0.662 | 0.658 |
| GTM | 0.695 | 0.680 |
| F1 | 0.699 | 0.709 |
| PREC | 0.708 | 0.745 |
| RECL | 0.690 | 0.676 |

Table 2: Official evaluation scores on eval09 obtained by primary Chinese-English baseline system vs. semi-supervised contrastive system, truecased version.

| System | BLEU | PER | Meteor | NIST |
|---|---|---|---|---|
| case+punc | 0.41 | 0.42 | 0.66 | 7.05 |
| no case+punc | 0.40 | 0.45 | 0.62 | 7.30 |

Table 3: Chinese-English translation results on the IWSLT09 eval set - subset of the official evaluation scores.

previous years, several techniques that were observed to increase MT performance (e.g. POS-based language models) did not show improvements on this years tasks, possibly due to the limitation of only using the BTEC training data. Our main goal this year was the integration of novel semi-supervised learning techniques, in particular semi-supervised ranking. Results from this method are inconclusive, improving some performance measures while decreasing others. This is in contrast to earlier experiments on two previous IWSLT data sets – further experiments need to be conducted to determine the reason for this discrepancy.

## 9. Acknowledgements

This work was partially funded by NSF grant IIS-0840461.

## 10. References

[1] Bennett, K., Demiriz, A., and Maclin, R., "Exploiting Unlabeled Data in Ensemble Methods", *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[2] Chang P-C., M. Gally and C. Manning, "Optimizing Chinese Word Segmentation for Machine Translation Performance", *ACL 2008 Third Workhop on Statistical Machine Translation*, 2008.

[3] Freund, Y., Iyer, R., Schapire, R., and Singer, Y., "An Efficient Boosting Algorithm for Combining Preferences", *Journal of Machine Learning Research 4*, 2003.

| System | BLEU | PER | Meteor | NIST |
|---|---|---|---|---|
| case+punc | 0.48 | 0.35 | 0.72 | 6.85 |
| no case+punc | 0.48 | 0.38 | 0.69 | 6.93 |

Table 4: Arabic-English translation results on the IWSLT09 eval set - subset of the official evaluation scores.

[4] Stolcke, A., "SRILM: An Extensible Language Modeling Toolkit", *Proceedings of ICSLP*, 2002.

[5] Habash, N. and O. Rambow, "Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop", *Proceedings of ACL*, 2005

[6] Byrne, W., and Deng, Y., "MTTK: An Alignment Toolkit for Statistical MAchine Translation", *Proceedings of HTL/NAACL*, New York, 2006.

[7] Koehn, P. and Och, F.J. and Marcu, D., "Statistical phrase-based translation", *Proceedings of HLT/NAACL*, Edmonton, Canada, 2003.

[8] Koehn, P. et al., "Moses: Open Source Toolkit for Statistical Machine Translation", *Proceedings of ACL demo session*, Prague, 2007.

[6] Och, F.J., and Ney, H., "A systematic comparison of various statistical alignment models", *Computational Linguistics 29(1)*, 19-52, 2003

[9] Och, F.J., "Minimum Error Rate Training for Statistical Machine Translation", *Proceedings of ACL*, Sapporo, Japan, 2003.

[10] Ratnaparkhi, A., "A maximum entropy part-of-speech tagger", in *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, 1996.

[11] Zhang, B. and J.G. Kahn, *Evaluation of Decatur Text Normalizer for Language Model Training*, Technical report, University of Washington