# The MIT-LL/AFRL IWSLT-2009 MT System

*Wade Shen, Brian Delaney, A. Ryan Aminzadeh*†

MIT Lincoln Laboratory
Information Systems and Technology Group
244 Wood St.
Lexington, MA 02420, USA
{swade,bdelaney,ryan.aminzadeh}@ll.mit.edu

*Tim Anderson, Ray Slyh*

Air Force Research Laboratory
Human Effectiveness Directorate
2255 H St.
Wright-Patterson AFB, OH 45433
{Timothy.Anderson,Raymond.Slyh}@wpafb.af.mil

## Abstract

This paper describes the MIT-LL/AFRL statistical MT system and the improvements that were developed during the IWSLT 2009 evaluation campaign. As part of these efforts, we experimented with a number of extensions to the standard phrase-based model that improve performance on the Arabic and Turkish to English translation tasks.

We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2008 system, and experiments we ran during the IWSLT-2009 evaluation. Specifically, we focus on 1) Cross-domain translation using MAP adaptation and unsupervised training, 2) Turkish morphological processing and translation, 3) improved Arabic morphology for MT preprocessing, and 4) system combination methods for machine translation.

## 1. Introduction

During the evaluation campaign for the 2009 International Workshop on Spoken Language Translation (IWSLT-2009) our experimental efforts centered on 1) improved statistical modeling for phrase-based MT, specifically, better modeling for sparse data, and 2) experiments with system combination.

In this paper we describe improvements over our 2008 baseline systems and methods we used to combine outputs from multiple systems. For a more full description of the 2008 baseline system, refer to [1].

The remainder of this paper is structured as follows. In section 2, we present an overview of our baseline system and the minor improvements to this standard statistical MT architecture that we incorporate. In sections 3, 4, 5, 6 and 7 we describe experiments for cross-domain adaptation, better Turkish and Arabic morphological processing, improved handling of speech input and our implementation of MT system combination. Section 8 describes the systems we submitted for this year's evaluation and their results.

†This work is sponsored by the Air Force Research Laboratory under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

### 1.1. IWSLT-2009 Data Usage

We submitted systems for Turkish-to-English and Arabic-to-English language pairs. In each case, we used data supplied by the evaluation for each language pair for training and optimization.

For cross-domain adaptation experiments we trained initial models using the ISI Arabic-English Automatically Extracted Parallel Corpus [3]. The IWSLT training data was used to adapt these initial models to the IWSLT domain. As these models make use of non-IWSLT data, they were not submitted for official evaluation.

We employ a minimum error rate training process to optimize model parameters with a held-out development set. The resulting models and optimization parameters can then be applied to test data during decoding and rescoring phases of the translation process.

## 2. Baseline System

Our baseline system implements a fairly standard SMT architecture allowing for training of a variety of word alignment types and rescoring models. It has been applied successfully to a number of different translation tasks in prior work, including prior IWSLT evaluations. The training/decoding procedure for our system is outlined in Table 1. Details of the training procedure are described in [4].

### 2.1. Phrase Table Training

To maximize phrase table coverage, we combine multiple word alignment strategies, extending the method described in [5]. For all language pairs, we combine alignments from IBM model 5 (see [8] and [9]) with alignments extracted using the competitive linking algorithm (CLA) described in [6] and the Berkeley Aligner [7]. Phrases were extracted from both types of alignments and combined in one phrase table. This was done by summing counts of phrases extracted from alignment types before computing the relative frequencies used in the our phrase tables.

| Training Process | |
|---|---|
| 1. | Segment training corpus |
| 2. | Compute GIZA++, Berkeley and Competitive Linking Alignments (CLA) for segmented data [5] [6] [7] |
| 3. | Extract phrases for all variants of the training corpus |
| 4. | Split word-segmented phrases into characters |
| 5. | Combine phrase counts and normalize |
| 6. | Train language models from the training corpus |
| 7. | Train TrueCase models |
| 8. | Train source language repunctuation models |
| **Decoding/Rescoring Process** | |
| 1. | Decode input sentences use base models |
| 2. | Add rescoring features (e.g. IBM model-1 score, etc.) |
| 3. | Merge N-best lists (if input is ASR n-best) |
| 4. | Rerank N-best list entries |

Table 1: *Training/decoding structure*

| Decoding Features |
|---|
| $P(\mathbf{f}\|\mathbf{e})$ |
| $P(\mathbf{e}\|\mathbf{f})$ |
| $LexW(\mathbf{f}\|\mathbf{e})$ |
| $LexW(\mathbf{e}\|\mathbf{f})$ |
| Phrase Penalty |
| Lexical Backoff |
| Word Penalty |
| Distortion |
| $\hat{P}(\mathbf{E})$ – 4-gram language model |
| **Rescoring Features** |
| $\hat{P}_{rescore}(\mathbf{E})$ – 5-gram LM |
| $\hat{P}_{class}(\mathbf{E})$ – 7-gram class-based LM |
| $P_{Model1}(\mathbf{F}\|\mathbf{E})$ – IBM model 1 translation probabilities |

Table 2: *Independent models used in log-linear combination*

### 2.2. Language Model Training

During the training process we built n-gram language models for use in decoding/rescoring, TrueCasing and repunctuation. In all cases, the SRI Language Modeling Toolkit [10] was used to create interpolated Knesser-Ney LMs. Additional class-based language model were also trained for rescoring. Some systems made use of 3- and 7-gram language models for rescoring trained on the target side of the parallel text.

### 2.3. Optimization, Decoding, and Rescoring

Our translation model assumes a log-linear combination of phrase translation models, language models, etc.

$$\log P(\mathbf{E}|\mathbf{F}) \propto \sum_{\forall r} \lambda_r h_r(\mathbf{E},\mathbf{F})$$

To optimize system performance we train scaling factors, $\lambda_r$, for both decoding and rescoring features so as to minimize an objective error criterion. This is done using a standard Powell-like grid search using a development set [11].

A full list of the independent model parameters that we used in our baseline system is shown in Table 2. All systems generated N-best lists that are then rescored and reranked using either a MAP or an MBR (Minimum Bayes Risk) criterion.

These model parameters are similar to those used by other phrase-based systems. For IWSLT, we also add a source-target word translation pairs to the phrase table that would not have been extracted by the standard phrase extraction heuristic from IBM model 5 word alignments. These phrases have an additional lexical backoff penalty that is optimized during minimum error rate training.

This system serves as the basis for a number of the contrastive systems submitted during this year's evaluation. Contrastive systems differ in terms of their rescoring configuration (e.g. language models, MBR) and the data used to train them (some system made use of additional lexicon data). Each of the contrastive systems was used as a component for system combination. The combined output for each of the Turkish-to-English and Arabic-to-English tasks was submitted as our primary system. Detailed differences of each submitted system can be found in section 9.

The `moses` decoder [12] was used for our baseline system and for confusion network decoding. Additionally, we also used an FST-based decoder developed for speech input applications.

## 3. Cross Domain Adaptation

During this evaluation we explored methods to adapt general-purpose phrase-based models to the IWSLT (travel domain) task. To this end, we built a general purpose model in Arabic using training data from the ISI automatically extracted parallel corpus [3]. These models were trained using over 500k sentence pairs of newswire data. Using the provided training data from the IWSLT evaluation, we explored two approaches to adapt the model to the IWSLT task:

1. **Semisupervised Adaptation:** We use the general-purpose model to generate translations of IWSLT source data. Machine translated sentences that are deemed to be "high quality" are then chosen for adaptation. Adaptation is performed using the selected source sentences from IWSLT and their corresponding MT output (i.e. no reference data is used for adaptation).

2. **Human-in-the-loop Adaptation:** We employ the general-purpose model to generate translations of the IWSLT Arabic source data. Then we select sentence pairs (both source and target) of the training set deemed to have "poor translation quality" (see below) to create an adaptation set. The resulting adaptation set is then used to update both the phrase table and language models. This simulates an active learning paradigm in which source sentences with poor output

from the MT system are sent to a translator for correction and then used for retraining.

### 3.1. Translation Quality Scoring

With both methods described above, we assume the presence of a human judge who can assign quality ratings to MT output. Due to cost and resource limitations, we used METEOR scores as a proxy for human judgement as it has been shown to correlate well [13]. Figure 1 shows the distribution of scores when decoding the IWSLT Arabic training set using the general-purpose model. Higher scoring sentences are the focus of the semisupervised adaptation approach whereas the human-in-the-loop method uses lower-scoring sentences as candidates for human translation.

### 3.2. Adaptation Methods

Given an initial (general-purpose) model and a small amount of in-domain data, we attempt to adapt the general-purpose model to the domain of interest. We explored three approaches for adaptation:

1. *Language Model Adaptation* In this case, we use a small set of target sentences from the domain of interest to generate a new language model. This language model is interpolated with a general purpose language model so as to maximize BLEU score. In the semisupervised version of this procedure, target language sentences are generated by decoding source language sentences selected from the IWSLT training set. In the human-in-the-loop condition, target sentences are chosen from the IWSLT training bitext.

2. *Phrase Table Adaptation* We apply MAP adaptation based on the approach described by Bacchiani et al [25] for language models. This approach interpolates phrase probabilities from both the general purpose phrase table and a phrase table trained on the adaptation data as described below:

$$\hat{p}(s|t) = \lambda p_{iwslt}(s|t) + (1-\lambda)p_{gp}(s|t) \qquad (1)$$

   where $p_{gp}$ and $p_{iwslt}$ are phrase probability estimates from the general purpose and IWSLT-domain models respectively, and $\lambda$

$$\lambda = \frac{N_{iwslt}(s,t)}{N_{iwslt}(s,t)+\tau} \qquad (2)$$

   where $\tau$ is the MAP relevance factor and $N_{iwslt}(s,t)$ is the observed count of phrase pair $(s,t)$. We set $\tau$ to a fixed value for all experiments.

   Phrases that do not exist in either the IWSLT training set or the general purpose phrase table are assumed to have probability zero in equation 1.

3. *Combined PT and LM adaptation*

## 4. Turkish Preprocessing

Turkish is an agglutinative language with a rich derivational and inflectional morphology. Many Turkish words are formed from the application of suffixes to a relatively small set of core noun and verb forms. This results in a potentially large vocabulary size and poor probability estimates when aligning Turkish-English parallel texts. We applied a rule-based Turkish morphological analyzer [14] to the Turkish texts and split morphemes into individual tokens. When taken in isolation, many morphological breakdowns of surface forms are ambiguous without the context of surrounding words. However, we achieved the best performance simply by choosing the first morphological parse for each surface form.

## 5. Count-Mediated Morphological Analysis for Arabic

Arabic is a morphologically rich language [15, 16], and various work (*e.g.,* as described in [17, 18]) has indicated that it can be advantageous to separate surface tokens into their morphological constituents for machine translation. In our system for IWSLT 2007, we employed a light morphological analysis procedure we called AP5 [2], and in our system for IWSLT 2008, we added a preprocessing step to AP5 to remove various diacritics [1]. In our 2009 system, AP5 (with the diacritic removal process) was again central to our Arabic-English system; however, our primary system was a combination of a number of subsystems, and several of these subsystems employed a modification to the AP5 process that we call Count-Mediated Morphological Analysis (CoMMA).

As described in [17, 18], Arabic has three possible levels of clitics that can be attached to a base form in a strict order:

```
[CONJ+ [PART+ [Al+ BASE +PRON]]]
```

A base may have the definite article `Al+` or a pronoun (`+PRON`) but not both. Particles (`PART+`) include `l+` "to/for", `b+` "by/with", and `k+` "as/such". (We do not segment the particle `s+` "will/future tense" in AP5.) Finally, the possible attached conjunctions (`CONJ+`) include `w+` "and" and `f+` "so". In [17], it was shown that the degree (*i.e.,* level) of clitic segmentation that performs best for statistical machine translation depends on the amount of training data available. With small amounts of training data, one should segment all of the clitics from the base; with larger amounts of training data, fewer levels should be split off from the base. In [18], it was further shown that various processing schemes that segment different levels of clitics could be combined to improve performance in a decoding-plus-rescoring method of combination.

Based on the insights of [17, 18], we also investigated multiple ways of segmenting the input Arabic text, although by a different method. Rather than segmenting off the same level of clitics for all surface tokens, where the "best" level depends on the overall amount of training data, the CoMMA process segments all levels of clitics (with AP5) for each token that occurs in the training data fewer times than a user-chosen threshold. Tokens that occur at least as many times as the threshold are passed through to the output unsegmented. Note that all tokens have diacritics removed before any part of the CoMMA process is applied. Any morphological analyzer (especially a light analyzer such as AP5) is bound to make mistakes for some tokens, and consistent mistakes made on tokens that occur frequently will in turn be frequent, whereas mistakes made on a token that occurs infrequently will in turn be infrequent.

## 6. Improved Speech Translation

### 6.1. Finite State Transducer System

We have successfully implemented a phrase-based translation system capable of directly translating ASR lattices via finite state transducers. Finite state transducers (FSTs) provide a useful framework for natural language processing applications as the implementation details of graph optimization and search are handled through a software library that operates on a common state machine representation. A detailed explanation our of FST system can be found in [2] and [1]. The basic idea is outlined below.

Our decoder takes as input a finite state acceptor which represents either a single input sentence or the output from a speech recognizer. The best translation can be described as the best path through the transducer given by:

$$E = I \circ P \circ D \circ T \circ L \tag{3}$$

where $\circ$ represents the composition operation. $I$ represents the input acceptor and $P$ is the phrase segmentation transducer. The transducer given by $D$ performs phrase permutations. $T$ and $L$ are the translation and language models, respectively.

We have made significant improvements to the pruning algorithm used in the FST decoder. Previous versions of the decoder relied on a generic Viterbi beam search routine common to many FST toolkits which did not account for optimizations specific to machine translation. In the FST decoding algorithm, beam search is used during the final step in (3), where the language model transducer, $L$, is applied. This step results in a potentially large node expansion, and both beam and histogram pruning are required to control memory usage. The generic search only pruned the set of expanded nodes resulting from a single node in the input transducer. In machine translation terms, the search only pruned the set of possible n-gram expansions from each predecessor node in isolation. There was no global pruning, and all distortion and phrase segmentation paths remained in the final search graph. With minimal changes to the search algorithm, we were able to implement phrase level pruning, much like that used in the Moses decoder.

First, input phrase boundaries were marked with a special # character that results in a null output, much like $\epsilon$. This serves two purposes: 1) determinization can be used to greatly reduce the size of the input phrase transducer after phrase swapping and 2) the search algorithm can use the special character to key on phrase boundaries and trigger a global phrase level pruning routine. However, the input transducer must first be topologically sorted according to a modified breadth-first search routine, where tree depth refers number of source words covered. In this way, a global beam pruning routine can be triggered each time the # character is encountered, signaling a change in the number of covered source words.

The result of these improvements are faster decoding times and reduced memory usage for a given configuration. We are also able to apply the phrase swapping transducer up to three times for longer distance phrase reordering. Typically, this operation can only be done twice as the total number of resulting output paths uses too much memory. This was particularly helpful in the Turkish-English language pair, where the systems benefited from increased distortion limits.

## 7. System Combination

In order to take advantage of the strengths of our various modeling and decoding techniques, we employ a system combination technique similar to the one presented in [21]. This is based on the successful ROVER technique used in automatic speech recognition [22]. In ROVER, individual words are aligned to minimize edit distance, and confusion networks are generated from these alignments. A voting algorithm is used to select the best word sequence with the lowest expected word error rate. In speech recognition, this process is relatively straightforward given the strict word order defined by the acoustics.

In machine translation, the system combination problem is compounded by many possible phrase choices and word orderings between systems. To combat this problem, each system serves as the skeleton system once, and all other system outputs are aligned to it. Confusion networks are generated for each skeleton alignment and the union of all confusion networks is taken. This final union network is then scored to find the best output sentence. The advantage of this technique over simply selecting the best system output is that the effect of combination can be localized within segments.

In our implementation of this round-robin confusion network scheme, we have added some additional features including a language model, word penalty, and a prior probability on choosing a particular system as the skeleton. To further improve the combination, we use a weighted voting scheme. All of these feature weights are optimized on a held-out set using Nelder-Meade simplex optimization to maximize the BLEU score.

In order to form the confusion networks, we use alignments provided by the translation error rate (TER) scoring tool [23]. TER performs a string alignment allowing for word movement via a beam search. We have modified the beam search to include partial matching via wordnet synonymy or word stems. Synonyms across candidate systems are considered matches (e.g. "attorney" is equivalent to "lawyer".) This results in an improved set of alignments and better confusion networks.

Each alignment set is converted to a confusion network where skipped words are allowed via NULL arcs. Each individual word, $w_i$, forms an arc with a posterior probability equal to the normalized sum of all system weights, $\lambda_n$, that produced word $w_i$. NULL arc probabilities are also included in this calculation.

In the final weighted confusion network, the hypothesis score for word sequence $\mathcal{W}$ is given by:

$$\begin{aligned}\log(P_{\mathcal{W}}) &= \sum_{i=0}^{I_k}\left[\log\left(\sum_{n\in w_i}\frac{\lambda_n}{\sum_{l=0}^{N}\lambda_l}\right)\right] + \lambda_N Len(\mathcal{W}) \\ &+ \lambda_{N+1}\log(P_{LM}(\mathcal{W})) + \lambda_{N+2}\log(\beta_k) \qquad (4)\end{aligned}$$

where $I_k$ is the number of confusion pairs in the branch with system $k$ as the skeleton, $N$ is the total number of systems, and $\lambda_0$ through $\lambda_{N+2}$ are the weights optimized by a simplex minimization procedure. Note that (4) is not log-linear with respect to the system weights, $\lambda_n$. The main kernel contains the summation over all confusion sets of the log of the sum of weighted posteriors and is more easily optimized via non-gradient based methods. The system priors, $\beta_k$, are given for each system to discourage poorly performing systems from taking the role as the skeleton. For our system we used the normalized BLEU scores from a held-out data set as system priors. Additionally, each sentence output is assigned a word penalty based on the total number of words, $Len(\mathcal{W})$, so that the sentence length can be properly optimized. Finally, a language model, $P_{LM}(\mathcal{W})$ is applied to the output sequence. The language model helps to reject hypotheses due to improper alignments, such as repeated or missing words. This formulation is similar to the one presented in [24], but here we have added a separate prior probability for each system and the word posteriors are computed only with the normalized $\lambda_n$ system weights.

## 8. Experiments

With each of the enhancements presented in prior sections, we ran a number of development experiments in preparation for this year's evaluation. This section describes the development data that was used for each evaluation track and results comparing the aforementioned enhancements with our baseline system. Our experiments focused on the Turkish-to-English (BTEC) and Arabic-to-English (BTEC) tasks.

### 8.1. Development Data

Tables 3 describes the development and training set configurations used for each language pair in this year's evaluation.

For Turkish, development experiments were conducted using `dev1` for optimization and `dev2` for development testing and system combiner optimization. For Arabic, `dev6` and `dev7` were used for optimization and development testing respectively.

| | | Turkish | English |
|---|---|---|---|
| `train` | Sentences | 19,972 K | |
| | Running words | 142,2519 | 161,171 |
| | Avg. Sent. length | 7.14 | 8.07 |
| | Vocabulary | 17,085 | 6,766 |
| `dev1` | Sentences | 506 | |
| | Running words | 2,908 | 4,101 |
| | Avg. Sent. length | 5.89 | 8.11 |
| `dev2` | Sentences | 500 | |
| | Running words | 2,980 | 4,056 |
| | Avg. Sent. length | 5.82 | 8.11 |
| | | **Arabic** | **English** |
| `train` | Sentences | 19,972 | |
| | Running words | 130,650 | 161,171 |
| | Avg. Sent. length | 6.54 | 8.07 |
| | Vocabulary | 18,121 | 6,766 |
| `dev6` | Sentences | 489 | |
| | Running words | 2,388 | 3,082 |
| | Avg. Sent. length | 4.88 | 6.30 |
| `dev7` | Sentences | 507 | |
| | Running words | 3,224 | 3,461 |
| | Avg. Sent. length | 6.36 | 6.83 |

Table 3: *Corpus Statistics for All Language Pairs*

### 8.2. Baseline Experiments

Turkish and Arabic data sets were processed using the morphological analysis procedures described above. The resulting text was then used for training, optimization and decoding. Tables 4 and 5 show the performance of our baseline systems on development data with AP5 preprocessing and Bilkent's morphology for Arabic and Turkish respectively. The Arabic system shown in these tables vary in terms of whether they use lexical approximation [19] and in terms of the decoder (either `moses` or our FST-based decoder).

| *System* | `dev6` | `dev7` |
|---|---|---|
| Phrase-based FST decoder + lex-approx | 55.84 | 55.59 |
| Standard Phrase-based system (no lex-approx) | 55.07 | 56.20 |
| Standard Phrase-based system (w/ lex-approx) | 53.24 | 54.66 |

Table 4: *Arabic Baseline Systems*

### 8.3. Domain Adaptation Experiments

We conducted a number of experiments comparing language model and phrase table adaptation in the context of both semisupervised and human-in-the-loop methods. Because these experiments make use of data from outside the IWSLT

| *System* | dev1 | dev2 |
|---|---|---|
| Phrase-based FST decoder | 64.63 | 60.79 |
| Standard Phrase-based system | 66.49 | 62.91 |
| + drop unknown words | 67.50 | 63.53 |
| + optimize NIST + BLEU | 67.08 | 63.65 |

Table 5: *Turkish Baseline Systems*

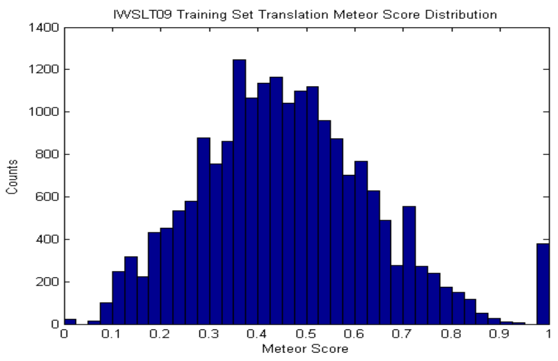

Figure 1: *Distribution of METEOR scores for sentences from the IWSLT training data*

evaluations, none of these systems were used for MIT/LL submission runs.

We evaluated the performance of both human-in-the-loop and semisupervised adaptation methods as a function of the adaptation data size. To this end, we decoded the IWSLT training data and ranked sentences by METEOR score (see figure 1). The data was then divided into octiles and each method was evaluated using fractions of this data between 1/8 and 8/8 for adaptation.

Figure 2 shows the results on dev7 when using an semisupervised method for adaptation. Using no adaptation, the general purpose models yield a BLEU score of 23.04 (significantly worse than the 56.20 achieved by our IWSLT baseline system). The red line shows the performance of an phrase table trained on the IWSLT-2009 training data using no reference data (also no general purpose phrase table or language model is used other than to decode the IWSLT-2009 training data set). The green and cyan lines show the performance of language model adaptation (without phrase table adaptation) and phrase table adaptation (without LM adaptation) respectively. The blue line shows the performance of joint phrase table and language model adaptation. In all cases, use of adaptation can outperform the baseline. Gains of 12-17% relative are possible when choosing the best performing sentences for adaptation of both the language model and phrase table (i.e. the lower octiles of the graph) with most of the improvement arising from language model adaptation. Even using all of the data, which does not require supervision (in terms of sentence quality judgements), results in a 9% relative improvement.

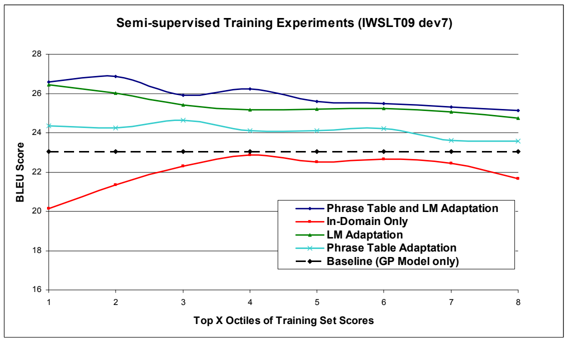

Figure 2: *Semisupervised Adaptation Results*

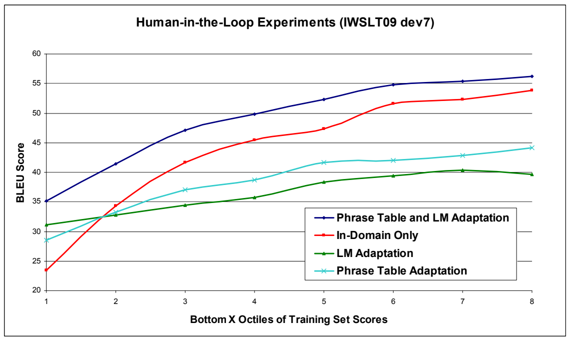

Figure 3: *Human-in-the-loop Adaptation Results*

Figure 3 shows the performance of human-in-the-loop adaptation as a function of adaptation data size. The red line indicates the performance of an IWSLT model trained on data from each of the adaptation data subsets (trained from scratch, no adaptation). This is a baseline making no use of general purpose models. The blue line shows the performance of human-in-the-loop adaptation using adapted phrase tables and language models while the green and cyan lines show language model adaptation (without phrase table adaptation) and phrase table adaptation (without language model adaptation) respectively. In all data sizes, adaptation (using both phrase table and language model approaches) outperforms the baseline significantly and interestingly, while both phrase table and language model adaptation methods improve performance in combination, their gains are complementary.

As one might expect, the relative improvement is greatest at smaller data sizes (using 1/8 of the IWSLT training data results in 49.8% relative improvement), but even when using all the IWSLT training data this adaptation method results in an improvement of 2.2 BLEU points (4.5% relative).

Table 6 summarizes these results.

## 8.4. Arabic Morphology Experiments

We evaluated the translation results from the CoMMA process at several threshold levels. We removed all diacritics from the training set and all of the development sets, and

| *System* | dev7 | eval |
|---|---|---|
| General Purpose Model Only | 23.06 | 21.35 |
| GP + Unsupervised Adaptation | 25.74 | 23.86 |
| GP + Semisupervised Adaptation (top $\frac{1}{4}th$) | 27.19 | 25.89 |
| IWSLT-only Baseline | 54.63 | 52.69 |
| GP Model + Human-in-the-Loop Adaptation | 56.57 | 56.11 |

Table 6: *Summary of Adaptation Experiment Results*

obtained token counts for all tokens in the Arabic side of the augmented training set consisting of the original training set and development sets one through five. For a given threshold, the CoMMA process was then applied (using the file of token counts) to the Arabic sides of the augmented training data and development sets six (dev6 ) and seven (dev7 ). All systems then used the baseline training, optimization, and Moses decoding process of our IWSLT 2008 Arabic-English system as outlined in [1] (without lexical approximation). Two experiments were run, one used dev6 for optimization with dev7 for testing, while the other used dev7 for optimization and dev6 for testing. After optimization, the best system in terms of BLEU score for TrueCased output was used for testing. Note that a CoMMA threshold of zero means that no token was segmented, while a threshold of 10,000 means that all tokens were segmented (as in the original AP5) as the only token to appear in the augmented training data more than 10,000 times was the period.

Table 7 shows the results in terms of mean BLEU scores for TrueCased output on dev6 and dev7 versus various CoMMA thresholds. For both data sets, all of the CoMMA thresholds 20 or higher resulted in substantially better performance than applying no morphological analysis (*i.e.,* at the CoMMA threshold of zero). For dev6 , the best CoMMA threshold was found to be 2,000, which performed 0.69 BLEU points better than segmenting all tokens (*i.e.,* at the CoMMA threshold of 10,000). However, for dev7 , all of the CoMMA thresholds 20 or higher resulted in similar performance. Despite the similar BLEU score performance on dev7 across CoMMA thresholds other than zero, the actual translations were often quite different.

| CoMMA | Mean BLEU | |
|---|---|---|
| Threshold | dev6 | dev7 |
| 0 | 50.00 | 51.94 |
| 20 | 53.92 | 54.29 |
| 200 | 53.14 | 54.64 |
| 2,000 | 54.02 | 54.57 |
| 10,000 | 53.33 | 54.48 |

Table 7: *Mean BLEU scores for CoMMA systems versus threshold.*

The results on dev7 as well as the system combination results of [18], suggested a further experiment in system combination with the CoMMA systems. Combining the CoMMA system at a threshold of 10,000 with the other three Arabic-English MT systems described in Sections 8.2, resulted in a BLEU score on lower case output of 56.58 for dev6 and 58.65 for dev7 . Combining the CoMMA systems at thresholds of 20, 200, 2,000, and 10,000 with the other three Arabic-English MT systems described in Sections 8.2, resulted in a BLEU score on lower case output of 57.08 for dev6 and 60.15 for dev7 . Thus, using all four of the CoMMA systems with thresholds other than zero yielded an additional 0.50 BLEU points on dev6 and 1.50 BLEU points on dev7 over using just the CoMMA system at 10,000 when combined with our other three Arabic-English MT systems. Thus, the CoMMA systems at thresholds of 20, 200, 2,000, and 10,000 were all used in this year's system.

## 9. Evaluation Summary

As part of this year's evaluation we experimented with cross-domain adaptation, Turkish morphological analysis, improved Arabic morphological processing and refinements to our multiple MT combination approach. These developments have helped to improve our system when compared with our 2008 baseline.

Table 8 summarizes each of the systems submitted for this year's evaluation.

## 10. Acknowledgments

We would also like to thank the staff of the Information Systems and Technology group at MIT Lincoln Lab for making machines available for this evaluation effort.

## 11. References

[1] Shen, W., Delaney, B., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2008 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Honolulu, HI, 2008.

[2] Shen, W., Delaney, B., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2007 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Trento, Italy, 2007.

[3] Munteanu, D. S. and Marcu, D., "ISI Arabic-English Automatically Extracted Parallel Text," Linguistic Data Consortium, Philadelphia, 2007.

[4] Shen, W., Delaney, B., and Anderson, T. "The MIT-LL/AFRL IWSLT-2006 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Kyoto, Japan, 2006.

[5] Chen, B. et al, "The ITC-irst SMT System for IWSLT-2005," In Proc. Of the International Workshop on Spoken Language Translation, Pittsburgh, PA, 2005.

[6] Melamed, D., "Models of Translational Equivalence among Words," In Computational Linguistics, vol. 26, no. 2, pp. 221-249, 2000.

| *Arabic-to-English Systems* | | |
|---|---|---|
| *System* | *Features* | BLEU |
| AE-primary | baseline with Arabic morphology | 57.17 |
| AE-primary + Tubitak | baseline combined tubitak systems [26] | 57.54 |
| *Turkish-to-English Systems* | | |
| *System* | *Features* | BLEU |
| TE-primary | baseline with Turkish morphology | 60.01 |
| TE-primary + Tubitak | baseline combined tubitak systems [26] | 60.71 |

Table 8: *Summary of Submitted Systems*

[7] Liang, P., Scar, B., and Klein, D., "Alignment by Agreement," Proceedings of Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL), 2006.

[8] Brown, P., Della Pietra, V., Della Pietra, S. and Mercer, R. "The Mathematics of Statistical Machine Translation: Parameter Estimation," Computational Linguistics 19(2):263–311, 1993.

[9] Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, I.D., Och, F.J., Purdy, D., Smith, N.A., Yarowsky, D., "Statistical machine translation: Final report," In Proceedings of the Summer Workshop on Language Engineering at JHU, Baltimore, MD 1999.

[10] Stolcke, A., "SRILM - An Extensible Language Modeling Toolkit," In Proceedings of the International Conference on Spoken Language Processing, Denver, CO, 2002.

[11] Och, F. J., "Minimum Error Rate Training for Statistical Machine Translation," In ACL 2003: Proc. of the Association for Computational Linguistics, Japan, Sapporo, 2003.

[12] Koehn, P., et al, "Moses: Open Source Toolkit for Statistical Machine Translation," Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic, June 2007.

[13] A. Lavie et al, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgment," In Proc. of the ACL, 2005.

[14] K. Oflazer and I. Kuruoz, "Tagging and morphological disambiguation of Turkish text," In Proceedings of the 4th Conference on Applied Natural Language Processing, Stuttgart, Germany, 1994.

[15] Badawi, E., Carter, M. G., and Gully, A., "Modern Written Arabic: A Comprehensive Grammar," Routledge: London, 2004.

[16] Mace, J., "Arabic Grammar: A Reference Guide," Edinburgh University Press: Edinburgh, 1998.

[17] Habash, N., and Sadat, F., "Arabic preprocessing schemes for machine translation," in *Proceedings of HLT-NAACL 2006,* (New York NY), June 2006.

[18] F. Sadat and N. Habash, "Combination of Arabic preprocessing schemes for statistical machine translation," in *Proceedings of COLING-ACL,* (Sydney, Australia), July 2006.

[19] Mermer, C., Kaya, H., and Dogan, M.U. "The TUBITAK-UEKAE Statistical Machine Translation System for IWSLT 2007," In Proc. of IWSLT, 2007.

[20] Besacier, L., Mahdhaoui, A., and Le, V.B., "The LIG Arabic/English speech translation system at IWSLT07," In Proc. of IWSLT, 2007.

[21] Matusov, E. and Ueffing, N. and Ney, H., "Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment," In Proc. of EACL, 2006.

[22] Fiscus, JG, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," In Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, 1997.

[23] Snover, M. and Dorr, B. and Schwartz, R. and Micciulla, L. and Makhoul, J., "A study of translation edit rate with targeted human annotation," In Proc. of AMTA, 2006.

[24] Rosti, A.V.I. and Matsoukas, S. and Schwartz, R., "Improved Word-Level System Combination for Machine Translation," In Proc. of ACL, 2006.

[25] M. Bacchiani and B. Roark, "Unsupervised Language Model Adaptation," In Proc. of ICASSP, 2003.

[26] Mermer, C., Kaya, H., and Dogan, M.U. "The TUBITAK-UEKAE Statistical Machine Translation System for IWSLT 2009," In Proc. of IWSLT, 2009.