The MIT-LL/AFRL IWSLT-2008 MT System

Wade Shen, Brian Delaney†

MIT Lincoln Laboratory
Information Systems and Technology Group
244 Wood St.
Lexington, MA 02420, USA
{swade,bdelaney}@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 2008 evaluation campaign. As part of these efforts, we experimented with a number of extensions to the standard phrase-based model that improve performance for both text and speech-based translation on Chinese and Arabic translation tasks.

We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2007 system, and experiments we ran during the IWSLT-2008 evaluation. Specifically, we focus on 1) novel segmentation models for phrase-based MT, 2) improved lattice and confusion network decoding of speech input, 3) improved Arabic morphology for MT preprocessing, and 4) system combination methods for machine translation.

1. Introduction

During the evaluation campaign for the 2008 International Workshop on Spoken Language Translation (IWSLT-2008) 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 2007 baseline systems and methods we used to combine outputs from multiple systems. For a more full description of the 2007 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 improved statistical modeling of phrases using segmentation probabilities, better 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-2008 Data Usage

We submitted systems for Chinese-to-English (Challenge Task), English-to-Chinese 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 the Chinese-to-English task, some of our systems made use of lexicon data from CEDICT [2] as parallel training data. These data are used to extract word/character alignments which are then expanded using slightly modified versions of standard heuristics. Phrases are extracted and counted, and the resulting phrase table is then used for decoding and rescoring. Language models are trained using the English side of each language pair, and some systems made use of a rescoring language model trained with LDC English Gigaword Corpus [3] and ISI's automatically extracted parallel corpus (Chinese-English) [4] with vocabulary limited to the training set. This process is described in detail in section 2.

Using the supplied development bitexts, 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 [5].

2.1. Phrase Table Training

To maximize phrase table coverage, we combine multiple word and character alignment strategies, extending the

Training Process
1. Word and character segment (Chinese-only) training corpus
2. Compute GIZA++, Berkeley and Competitive Linking Alignments (CLA) for segmented data [6] [7] [8]
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*

method described in [6]. For all language pairs, we combine alignments from IBM model 5 (see [9] and [10]) with alignments extracted using the competitive linking algorithm (CLA) described in [7] and the Berkeley Aligner [8]. 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.

Additionally, for Chinese-to-English translation, both word and character segmentation were used for training CLA, Berkeley and GIZA alignment models. Phrases were then extracted from all six alignments and combined. Word segmented phrases were resegmented into characters before counting.

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 [11] 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 that were trained with the English Gigaword corpus.

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, λ_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 [12].

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.

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*

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 Chinese-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 [13] was used for our baseline system and for confusion network decoding. Two other decoders were also used: 1) a direct-lattice decoder (used for ASR input in Arabic and Chinese) and 2) an internally developed phrase-based decoder that supports forced-alignment (used for systems that use segmentation models).

3. Phrase Segmentation Models

During this evaluation we developed improved segmentation models that allow for better scoring of phrases during decoding. Consider the following phrase-based model for the translation of a source sentence $\mathbf{F}$ to a target sentence $\mathbf{E}$:

$$P(\mathbf{E}|\mathbf{F}) \propto P(\mathbf{E}) * P(\mathbf{F}|\mathbf{E}) \quad (1)$$

$$\approx P(\mathbf{E}) * \max_{\substack{(\mathbf{f},\mathbf{e})_1^k \\ \in seg(\mathbf{F},\mathbf{E})}} p((\mathbf{f},\mathbf{e})_1^k) * \prod_{i=1}^{k} p(\mathbf{f_i}|\mathbf{e_i}) \quad (2)$$

where $seg(\mathbf{F}, \mathbf{E})$ denotes the set of possible segmentations of sentences $\mathbf{F}$ and $\mathbf{E}$ and a single segmentation $(\mathbf{f}, \mathbf{e})_1^k$ can be decomposed into phrase pairs $(\mathbf{f_i}, \mathbf{e_i})$ for $i = 1..k$. In the standard model the segmentation probability $p((\mathbf{f}, \mathbf{e})_1^k)$ is

assumed to be uniform across all possible segmentations and target sentences; as such, the model simplifies to the standard phrase-based model:

$$P(\mathbf{E}|\mathbf{F}) \approx P(\mathbf{E}) * \max_{(\mathbf{f},\mathbf{e})_1^k \in seg(\mathbf{F},\mathbf{E})} \prod_{i=1}^{k} p(\mathbf{f_i}|\mathbf{e_i}) \quad (3)$$

During decoding, all possible translations $\mathbf{E}$ and segmentations $(\mathbf{f},\mathbf{e})_1^k$ are jointly searched to find:

$$\mathbf{E}* = \arg\max_{\mathbf{E}} P(\mathbf{E}) * P(\mathbf{E}|\mathbf{F}) \quad (4)$$

The assumption of uniform segmentation leads to overestimation of likelihoods for paths that use longer phrases. To rectify this, most systems incorporate additional features such as phrase penalties and lexical translation probabilities in a log-linear model. A typical configuration (one used by our baseline system) that makes use of these features is shown below:

$$P(\mathbf{E}|\mathbf{F}) \propto P(\mathbf{F}|\mathbf{E})_1^{\lambda} * LexW(\mathbf{F}|\mathbf{E})_2^{\lambda} * exp(k)_3^{\lambda} \ldots \quad (5)$$

where k denotes the number of phrases in the segmentation that was used to compute $P(\mathbf{E}|\mathbf{F})$.

We propose to extend the standard model with a non-uniform model of phrase segmentation. Instead, we assume that the segmentation of each phrase is independent, such that:

$$P((\mathbf{f},\mathbf{e})_1^k) \approx \prod_{i=1}^{k} p(\mathbf{f_i}|\mathbf{F}) * p(\mathbf{e_i}|\mathbf{E}) \quad (6)$$

where the phrase segmentation probabilities $p(\mathbf{f_i}|\mathbf{F})$ and $p(\mathbf{e_i}|\mathbf{E})$ are modeled as:

$$p(\mathbf{f_i}|\mathbf{F}) \approx \frac{E_{\mathcal{F}}(\mathbf{f_i}|\lambda)}{N_{\mathcal{F}}(\mathbf{f_i})} \quad (7)$$

$$p(\mathbf{e_i}|\mathbf{E}) \approx \frac{E_{\mathcal{E}}(\mathbf{e_i}|\lambda)}{N_{\mathcal{E}}(\mathbf{e_i})} \quad (8)$$

where $\mathcal{F}$ and $\mathcal{E}$ denote the training set for which numerator and denominator counts are collected.

Source and target segmentation probabilities are computed using the EM algorithm over the training data. Specifically, we employ a forced-alignment procedure to compute the expected number of times each phrase occurs in the training data. This process is shown in detail below in Table 3.

To support the alignment needed in step 3 of this procedure, we built a new phrase-based decoder that supports forced-alignment of a source sentence to a supplied reference translation. Because of the large number of sentences that need to be aligned, it is critical that the decoder be implemented with maximum efficiency. Through efficient pruning, our decoder is able to align IWSLT sentences in < 2.5s (average) of processing per sentence with unlimited distortion.

Special handling of unknown words in the training data is needed to ensure that training sentences can be properly aligned. We allow unknown words in the source sentence to align to all possible target words, but with a heavy penalty. This forces target words that are legitimate translations of words in the source sentence to be preferred during the alignment process.

1.	Train standard phrase-based model
2.	Augment phrase model probabilities with initial segmentation probabilities
3.	Force align training bitexts and dump lattices
4.	Compute phrase-pair expected values using fixed λs from lattices (E-step)
5a.	Reestimate segmentation probabilities using equations 7 and 8 (M-step)
5b.	MER training to optimize model exponents (λs)
6.	Repeat 2-5

Table 3: *Phrase Segmentation Training Procedure*

Two submitted contrast systems make use of segmentation probabilities for text input decoding in the Arabic and Chinese tasks respectively: AE-constrast3 and CE-constrast3. Due to time constraints, these systems were trained with one iteration of EM training and rescored without additional language models.

4. Arabic Preprocessing

Arabic is a morphologically rich language [14, 15], and various work (as described in [16]) 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 [1]. We again used this procedure; however, we first applied additional text normalization to remove various diacritics. Normally, Modern Standard Arabic is written without short vowels and many other diacritics; however, some of the training and development sets have unusually large numbers of short vowels and diacritics present, which can lead to data sparsity during statistical training. These diacritics can also have negative consequences for our AP5 morphological analysis procedure. To help mitigate these negative effects, we investigated the removal of short vowels, shadda (which denotes consonant gemination), sukuun (which indicates the absence of a vowel), tanween (which mark grammatical cases), and tatweel (used to stretch letters in Arabic typography and which has no grammatical or semantic meaning).

5. Lexical Approximation for Arabic MT

Initial results on the Arabic-English task showed a higher rate of OOV words than expected. Rather than pass these words to the output, we chose to make a last-ditch effort at getting them correct through a lexical approximation technique similar to the one presented in [17]. Our version was implemented as a preprocessing step to the phrase table using a list

of OOV words. The top five known word candidates with a character edit distance less than a threshold from each OOV are chosen as possible translation candidates. All phrase table entries containing the translation candidates are replicated with the OOV word in its place. Although this simplistic approach uses no morphological information, it yielded a gain of 1.95 BLEU on `dev6` during our development testing and was used on several of our final component systems.

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 [1].

For the Chinese-English task, we used the FST system only on the ASR input condition. The first system, *CE-constrast7* used only the supplied 20k training data and devsets 1, 2, 4, 5, and 6 with one reference. Alignments from GIZA++, Berkeley Aligner, and Competitive Linking were derived from three different Chinese segmentation variants. We used the supplied segmentation, a resegmentation from Lingpipe, and a character segmented version of the data. All phrases were character segmented before counting. Punctuation was added to the input speech lattices wherever a word/punctuation bigram existed in the phrase table and the resulting lattice was rescored with a weighted punctuation language model. The FST system used a 4-gram language model during decoding, 5-gram and class 6-gram language models and IBM model1 scores during N-best reranking. The final Chinese-English system, *CE-contrast6* differs from the above with the addition of parallel text from the LDC Chinese-English dictionary.

For Arabic-English, we used the FST system for both the ASR and CRR input conditions. Only one system was created for Arabic-English, *AE-contrast1*. Similar to the Chinese condition, we used only the supplied parallel text and devsets 1, 2, and 4 with one reference. Phrases were extracted from three different alignment algorithms: GIZA++, Berkeley Aligner, and Competitive Linking. Input lattices were repunctuated as in the Chinese-English condition, and a 4-gram language model was used during decoding followed by 5-gram, class 6-gram, and IBM model1 N-best reranking. Preprocessing of the Arabic data consisted of removal of diacritics followed by AP5 normalization. The phrase table was augmented via lexical approximation (section 5) to reduce the OOV rate.

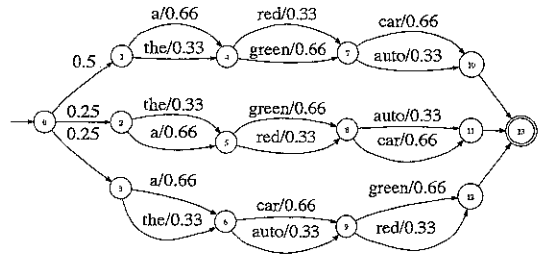

Figure 1: *System combination confusion network example*

6.2. Confusion Network Decoding

We applied the confusion network decoding strategy described in [1]. During the 2007 evaluation, it was noted that the multiword splitting method implemented in the SRILM `lattice-tool` was suboptimal. During this evaluation, we preprocessed character segmented lattices (Chinese) and morphologically preprocessed latttices (Arabic) using the splitting algorithm described in [18].

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 [19]. This is based on the successful ROVER technique used in automatic speech recognition [20]. 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 [21]. TER performs a string alignment allowing for word movement via a beam search. 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, λ_n, that produced word w_i. NULL arc probabilities are also included in this calculation.

Figure 1 illustrates this process for the following fictitious system outputs: "a red car," "the green auto," "a car green." The probabilities, assuming equal system weights and not including language model and word penalties, are shown on the arcs. The initial arcs from state zero contain the system prior probabilities for each skeleton alignment. The highest probability path through this network produces the sentence "a green car."

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

$$\begin{aligned}\log(P_{\mathcal{W}}) &= \sum_{i=0}^{I_k}\left[\log\left(\sum_{n\in w_i}\frac{\lambda_n}{\sum_{l=0}^{N}\lambda_l}\right)\right] + \lambda_N Len(\mathcal{W}) \\ &+ \lambda_{N+1}\log(P_{LM}(\mathcal{W})) + \lambda_{N+2}\log(\beta_k) \quad (9)\end{aligned}$$

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 λ_0 through λ_{N+2} are the weights optimized by a simplex minimization procedure. Note that (9) is not log-linear with respect to the system weights, λ_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, β_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 [22], but here we have added a separate prior probability for each system and the word posteriors are computed only with the normalized λ_n system weights.

8. Experiments

With each of the enhancements presented in prior sections, we ran a number of developement 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 Chinese-to-English (CT) and Arabic-to-English (BTEC) tasks[1].

[1]Unfortunately, due to resource constraints, similar experiments were not done for the English-to-Chinese (CT) task.

		Chinese	English
train	Sentences	40 K	
	Running words	148,219	161,171
	Avg. Sent. length	7.42	8.07
	Vocabulary	8,407	6,766
dev3	Sentences	506	
	Running words	3,209	3,271
	Avg. Sent. length	6.34	6.46
dev7	Sentences	246	
	Running words	1,305	1,540
	Avg. Sent. length	5.3	6.26
		Arabic	English
train	Sentences	19,972	
	Running words	130,650	161,171
	Avg. Sent. length	6.54	8.07
	Vocabulary	18,121	6,766
dev5	Sentences	500	
	Running words	4,652	6,332
	Avg. Sent. length	9.30	12.66
dev6	Sentences	489	
	Running words	2,388	3,082
	Avg. Sent. length	4.88	6.30
		English	Chinese
train	Sentences	40 K	
	Running words	161,171	148,219
	Avg. Sent. length	8.07	7.42
	Vocabulary	6,766	8,407
dev3	Sentences	506	
	Running words	3,273	3,209
	Avg. Sent. length	6.47	6.34
dev7	Sentences	246	
	Running words	1,321	1,305
	Avg. Sent. length	5.26	5.3

Table 4: *Corpus Statistics for All Language Pairs*

8.1. Development Data

Tables 4 describes the development and training set configurations used for each language pair in this year's evaluation.

8.2. Segmentation Model Experiments

Table 5 shows results of development experiments we ran in preparation for this evaluation (lower case, with punctuation). Each BLEU score represents the average of 10 optimization/rescore runs (optimized with dev3 or dev7 respectively). Though the gains are small, they are consistent despite only one EM iteration. More experiments are needed to refine and assess the performance of these models.

8.3. Arabic Morphology Experiments

Table 6 shows the results (lower case, with punctuation) of applying various levels of diacritic normalization as well as

System	dev7	dev3
Baseline (no rescore LMs)	39.6	52.9
+ phrase segmentation models	40.3	53.6

Table 5: *Segment EM results*

AP5 normalization to the data used in the Arabic-to-English task (averaged over 10 optimization/rescore runs). The AP5 normalization procedure used in our 2007 system removes the tanween characters. As such we examined the effect of other diacritics on MT performance. Comparing the baseline performance with the removal of all diacritics except the tanween (without any further AP5 processing), one can see that the diacritics other than the tanween have a dramatic effect on performance. Removing them yielded a mean score of 49.4, significantly better than 42.1, the score of our baseline with no preprocessing. Further removal of the tanween yielded approximately one additional BLEU point. Finally, removal of all diacritics (including the tanween) followed by the AP5 processing yielded an additional 3.2 BLEU points. All submitted Arabic systems removed all diacritics and then applied the AP5 processing.

Preprocessing Method	dev6
Baseline (No normalization or AP5)	42.06
Remove diacritics except tanween, no AP5	49.40
Remove all diacritics, no AP5	50.39
Remove all diacritics, apply AP5	53.55

Table 6: *A Comparison of Different Arabic Preprocessing Methods*

8.4. Speech Input Experiments

We conducted a number of development experiments to explore the performance of different speech decoding methods for both Arabic-English and Chinese-English translation. Tables 7 and 8 summarize the results of these experiments. All results are mean BLEU scores (lower-case with punctuation), averaged over 10 optimization/rescore runs. Note that we observe consistent gains by utilizing multiple hypotheses (i.e. lattice, confusion network and 20-best decoding).

ASR Decoding Method	dev3
1-Best	42.90
20-Best	46.13
Confusion Network	44.96

Table 7: *Comparison of Chinese-to-English ASR Decoding Methods*

ASR Decoding Method	dev5
1-Best	25.49
Confusion Network	26.78

Table 8: *Comparison of Arabic-to-English ASR Decoding Methods*

8.5. System Combination Experiments

For the Chinese-English task, we optimized our individual systems on the supplied Challenge Task development set. Using these optimized system weights, we produced lower-case 1-best output for all systems on the IWSLT08 evaluation set and dev3 , which was used to optimize the system combination weights. We chose dev3 because it used the same speech recognition parameter weights as the Challenge Task, but it was not clear if the speech output was produced by the same recognition system. We used the same devsets for our text input condition.

Due to a last minute bug in the word penalty optimization, the system combination favored shorter sentences. We chose to use only the seven longest reference sets from dev3 to encourage longer output and possibly lessen the impact of the brevity penalty. On the ASR system combination, we dropped the two systems which produced the shortest output (CE-contrast1 and CE-contrast5) to further reduce the possible impact of the brevity penalty.

The results for both the ASR and CRR conditions are shown in Table 9 and Table 10 (mixed-case with punctuation). The results shown for both the dev and eval sets are for BLEU with case and punctuation. For the ASR condition, we observed a gain of 2.37 BLEU over the best system output on the eval set. On the text condition, we lost 0.23 BLEU over the best system. This loss can possibly be attributed to the word penalty bug as well as the swapping of system ranking between the dev and eval sets. The best system on dev3 (CE-contrast4) was not the best system on the eval set (CE-contrast1), and therefore the optimizer produced weights that may have favored CE-contrast4 too much.

System Description	*Input*	dev3	eval
CE-contrast4	Conf. Net	45.80	31.93
CE-contrast3	1-Best	41.70	31.13
CE-contrast2	1-Best	41.65	31.41
CE-contrast7	Lattice	39.70	30.66
CE-contrast6	Lattice	38.84	31.02
Combination	–	–	*34.27*

Table 9: *System Combination Results for the Chinese-English ASR Input Condition*

For the Arabic-English task, we optimized our individual systems on the supplied dev6 and used dev5 to optimize the system combination parameters. As there was no appar-

System Description	dev3	eval
CE-contrast4	53.75	36.91
CE-contrast1	52.92	37.78
CE-contrast3	52.76	35.35
CE-contrast2	52.45	36.51
Combination	–	*37.55*

Table 10: *System Combination Results for the Chinese-English CRR Input Condition*

ent consistency across speech recognition systems used to produce the lattice outputs, we held constant the ASR parameters for those systems that optimize them specifically (i.e. our FST decoding system.) The same sets were used for the text input condition.

The results for the Arabic-English system combination are shown in Table 11 and Table 12 for the ASR and CRR conditions respectively (mixed-case, with punctuation). On the ASR condition, we achieved a significant gain of 3.29 BLEU, while the CRR condition yielded a gain of 1.44 BLEU.

On both the Arabic-English and Chinese-English data conditions, we noticed small gains by exploiting multiple ASR hypotheses through N-best lists, confusion networks, or lattices. However, the system combination yielded relatively large gains in both ASR conditions when combining translation outputs from these very different decoding input types. Each of these different speech translation systems produce complementary output that seems to combine well despite similar BLEU scores.

System Description	*Input*	dev5	eval
AE-contrast4	Conf. Net	25.69	45.31
AE-contrast3	1-Best	25.34	45.63
AE-contrast1	Lattice	24.53	44.49
AE-contrast2	1-Best	23.44	44.35
Combination	–	–	*48.92*

Table 11: *System Combination Results for the Arabic-English ASR Input Condition*

System Description	dev5	eval
AE-contrast4	27.95	55.07
AE-contrast3	27.91	54.91
AE-contrast1	26.03	50.81
AE-contrast2	28.25	51.79
Combination	–	*56.51*

Table 12: *System Combination Results for the Arabic-English CRR Input Condition*

9. Evaluation Summary

As part of this year's evaluation we experimented with novel phrase segmentation models, improved Arabic morphological processing and methods for combining multiple MT outputs. These developments have helped to improve our system when compared with our 2007 baseline.

Table 13 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-2007 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Trento, Italy, 2007.

[2] http://www.mandarintools.com/cedict.html

[3] Graff, D. "English Gigaword," Linguistic Data Consortium, Philadelphia, 2003.

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

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

Chinese-to-English Systems				
System	*CRR Features*	BLEU (CRR)	*ASR Features*	BLEU (ASR)
CE-primary	combined system	37.55	combined system	34.27
CE-contrast1	adds Gigaword 6g	37.78	1-best	31.49
CE-contrast2	adds Gigaword 6g + MBR	36.51	1-best + MBR	31.41
CE-contrast3	no rescoring LMs + segmentation models	35.35	1-best	31.13
CE-contrast4	adds Gigaword 3g LM + CEDICT	36.91	confusion network	31.93
CE-contrast5	adds Gigaword 3g LM + CEDICT	36.91	20-best	30.58
CE-contrast6	adds Gigaword 3g LM + CEDICT	36.91	FST + Lattice	31.02
CE-contrast7	adds Gigaword 3g LM + CEDICT	36.91	FST + Lattice	30.66
Arabic-to-English Systems				
System	*CRR Features*	BLEU (CRR)	*ASR Features*	BLEU (ASR)
AE-primary	combined system	56.51	combined system	48.92
AE-contrast1	FST	50.81	FST + lattice	44.49
AE-contrast2	baseline	51.79	1-best	44.35
AE-contrast3	no rescoring LM + segmentation models	54.91	1-best	45.63
AE-contrast4	adds Gigaword 3g LM	55.07	confusion network	45.31
English-to-Chinese Systems				
System	*CRR Features*	BLEU (CRR)	*ASR Features*	BLEU (ASR)
EC-primary	baseline	37.99	1-best	32.87

Table 13: *Summary of Submitted Systems*

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

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

[13] 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.

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

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

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

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

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

[19] 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.

[20] 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.

[21] 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.

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