REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
09-11-2010	Technical Paper	NOV 2010 - DEC 2010

4. TITLE AND SUBTITLE
The MIT-LL/AFRL IWSLT-2010 MT System

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Wade Shen, Tim Anderson, Ray Slyh, and A. Ryan Aminzadeh

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force
Air Force Research Laboratory
2255 H Street
Wright-Patterson AFB OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HECA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2008 system, and experiments we ran during the IWSLT-2010 evaluation. Specifically, we focus on 1) cross-domain translation using MAP adaptation, 2) Turkish morphological processing and translation, 3) improved Arabic morphology for MT preprocessing, and 4) system combination methods for machine translation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: U			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Zach Sweet
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
U	U	U	SAR	8	781-981-5997

The MIT-LL/AFRL IWSLT-2010 MT System

Wade Shen†
MIT/Lincoln Laboratory
Human Language Technology Group
244 Wood St.
Lexington, MA 02420, USA
swade@ll.mit.edu

Tim Anderson, Ray Slyh
Air Force Research Laboratory
2255 H St.
Wright-Patterson AFB, OH 45433
Timothy.Anderson@wpafb.af.mil
Raymond.Slyh@wpafb.af.mil

A. Ryan Aminzadeh
Department of Defense
ryan.aminzadeh@ugov.gov

Abstract

This paper describes the MIT-LL/AFRL statistical MT system and the improvements that were developed during the IWSLT 2010 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 also participated in the new French to English BTEC and English to French TALK tasks.

We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2008 system, and experiments we ran during the IWSLT-2010 evaluation. Specifically, we focus on 1) cross-domain translation using MAP adaptation, 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 2010 International Workshop on Spoken Language Translation (IWSLT-2010) 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 2009 baseline systems and methods we used to combine outputs from multiple systems. For a more full description of the 2009 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 developed. In sections 3, 4, 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-2010 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 [5] for AE tasks and the Europarl corpus for FE tasks. 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 [6].

2.1. Phrase Table Training

To maximize phrase table coverage, we combine multiple word alignment strategies, extending the method described in [7]. For all language pairs, we combine alignments from IBM model 5 (see [10] and [11]) with alignments extracted using the competitive linking algorithm (CLA) described in [8] and the Berkeley Aligner [9]. 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 [7] [8] [9]
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*

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 [12] was used to create interpolated Knesser-Ney LMs. Additional class-based language models 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, 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 [13].

In addition to the Powell-based approach, a number of our systems used the MIRA algorithm for weight optimization [22, 21, 23]. In this approach, weights are optimized subject to a maximum margin constraint in an online fashion. The equation below shows the update procedure for weights w_i corresponding to the ith online iteration of the algorithm.

$$\mathbf{w_i} = \mathbf{w_{i-1}} + \alpha * (\mathbf{h}(f, \hat{e}) - \mathbf{h}(f, e))$$

where $\hat{e}$ denotes the oracle translation for a source sentence f, $\mathbf{h}(f, e)$ is a vector of model scores corresponding to the translation of f into e, and α is an update scaling parameter defined as follows:

$$\alpha = max(0, min(C, \frac{\mathcal{L}(\hat{e}, e) - (s^{i-1}(f, \hat{e}) - s^{i-1}(f, e))}{||\mathbf{h}(f, \hat{e}) - \mathbf{h}(f, e)||}$$

$$s^{i-1}(f, e) = \mathbf{w_{i-1}} \cdot \mathbf{h}(f, e)$$

new

$\mathcal{L}(\hat{e}, e)$ defines a loss function (in our case, the BLEU score difference between the oracle translation, $\hat{e}$, and the current best translation, e. C is a limiter on the update scaling. It's easy to see that update size at each iteration is proportional to the difference between the loss value and the predicted score margin.

Weights $\mathbf{w_i}$ are updated sentence by sentence (order of presentation is randomized) until either a convergence criterion is met or a limit on the number of iterations is reached. Our implementation of MIRA follows the procedure in [22] for oracle selection and scoring.

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 ML 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*

These model parameters are similar to those used by other phrase-based systems. For IWSLT, we also add 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 [14] was used for our baseline system.

3. Cross Domain Adaptation

During this evaluation we re-examined the approach to cross domain adaptation that we presented in last year's evaluation [1]. To this end, we built a general purpose model in Arabic and French using training data from the ISI automatically extracted parallel corpus [5] and the Europarl corpus [4] for each language respectively. These models were trained using over 500k sentence pairs of newswire data. Using the provided training data from the IWSLT evaluation, we applied a variation of the MAP phrase table adaptation procedure described last year, which is shown in the equations below:

$$\begin{aligned} \hat{p}(s|t) &= \lambda p_{iwslt}(s|t) + (1-\lambda) p_{gp}(s|t) \\ \lambda &= \frac{N_{iwslt}(s,t)}{N_{iwslt}(s,t) + N_{gp}(s,t) + \tau} \end{aligned}$$

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

In this variation, the ratio of counts between $iwslt$ and gp models determines the weighting of the models. In last year's variation, lambda depends only on N_{iwslt} and if $N_{iwslt} >> \tau$ lambda approaches 1 (i.e. no adaptation).

As in last year's experiments, phrase table adaptation and language model interpolation were used jointly to improve performance. As these systems do not conform to the evaluation conditions, none of the submitted systems make use of this method.

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 [15] 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. Hamza Normalization for Arabic

Writers of Arabic sometimes adopt varying conventions regarding the use of the letter hamza with the letter alef. Some writers will place a hamza above an alef in situations where others would use only a bare alef (particularly with the definite article, "Al"). On the other hand, some writers will use a bare alef in situations that would call for an alef with a hamza above or below it. In our Arabic systems for IWSLT 2007–2009 [3, 2, 1], we employed a light morphological analysis procedure we called AP5, and this procedure accounted for some of these alef-hamza variations. At the beginning of a token, we normalized an alef with a hamza above or below it to a bare alef. After splitting a token into hypothesized morphemes, we normalized alef-hamza combinations at the beginning of morphemes to a bare alef. These normalizations improved our translation performance; however, they did not normalize all of the alef-hamza variations. This year, we experimented with normalizing all alef-hamza combinations (Unicode characters `\x{0623}` and `\x{0625}`) to bare alefs (Unicode `\x{0627}`) before applying any of the AP5 morphological processing, and this change improved the mean BLEU score from 54.15 to 54.96 on the IWSLT-postprocessed truecase output from the `dev7` data. As a result, we applied this global alef-hamza normalization as the first step in all of the Arabic subsystems used in our final submission.

6. Count-Mediated Morphological Analysis and Multi-Threshold Training

In our 2009 Arabic MT system [1], we employed a modification of our AP5 process that we called Count-Mediated Morphological Analysis (CoMMA). The CoMMA process segments only those tokens (with AP5) that occur 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. For this year's Arabic system, we again employed the CoMMA process, but with the global alef-hamza normalization discussed in section 5. We trained, optimized, and tested systems (on the `dev6` and `dev7` data) using CoMMA thresholds of 0, 20, 200, 2000, and 10,000. 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.

In our 2009 Turkish system, we used the Turkish morphological analyzer described in [15], but without any CoMMA process. For this year's Turkish system, we added the CoMMA process with the Turkish morphological analyzer of [15] in place of the AP5 Arabic analyzer. For Turkish, we considered thresholds of 0, 2, 20, 200, and 2,000. At a threshold of 2,000, all of the tokens that can be segmented by the morphological analyzer [15] are in fact segmented.

In addition to the standard CoMMA process for both Arabic and Turkish, we investigated the utility of a modification to the training process that we call CoMMA with Multi-Threshold Training (CoMMA-MTT). In the standard CoMMA process, a single threshold at a time is chosen, and the training, optimization, and testing data are all processed by CoMMA at the given threshold. With the CoMMA-MTT process, the source language training data are processed at all of the thresholds previously mentioned for that language, and the outputs are concatenated. The target (in this case, English) training data are replicated as many times as necessary to maintain parallel data. The alignment process is performed, and the phrase table is extracted. The development and testing data are then processed with a single threshold at a time. Thus, for the standard CoMMA process, the phrase tables are different for each threshold level, while for the CoMMA-MTT process, the phrase table is the same for different threshold levels. The development and testing data depend only on a single threshold.

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 [17]. This is based on the successful ROVER technique used in automatic speech recognition [18]. 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 [19]. 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, λ_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:

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

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 (1) 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 [20], 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 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. For French (BTEC), `dev2` was used for optimization and `dev3` was set aside for development testing. MT systems for the TALK task data used `dev1` for weight optimization and `dev2` as a held-out test set.

8.2. Baseline BTEC 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 decod-

		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
		French	English
train	Sentences	19,972	
	Running words	157,483	161,171
	Avg. Sent. length	7.89	8.07
	Vocabulary	8,739	6,766
dev2	Sentences	500	
	Running words	3,060	4,101
	Avg. Sent. length	6.05	8.11
dev3	Sentences	506	
	Running words	3,109	4,056
	Avg. Sent. length	6.21	8.11
		English	French
train	Sentences	83,923	
	Running words	877,531	840,776
	Avg. Sent. length	10.46	10.02
	Vocabulary	33,753	26,298
dev1	Sentences	787	
	Running words	7,425	7,476
	Avg. Sent. length	9.43	9.50
dev2	Sentences	520	
	Running words	5,087	5,076
	Avg. Sent. length	9.78	9.76

new

Table 3: *Corpus statistics for all language pairs*

ing. Tables 4 and 5 show the performance of our baseline systems on development data with AP5 preprocessing (with 2010 modifications) 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 [16], drop unknown words or make use of MBR as the scoring criterion. French preprocessing follows WMT specifications with additional splitting of contracted pronoun and preposition forms.

Arabic systems benefit from MBR rescoring, and both Arabic and French systems benefit from dropping of unknown words during decoding. MBR performance seems very sensitive to posterior scaling and N-best list size. As such, our default settings may not be optimal for MBR rescoring. Though lexical approximation didn't improve our baseline system, we found it beneficial to our final system combination.

System	dev6	dev7
Standard phrase-based system	56.16	56.22
Standard + MBR	56.51	56.20
+ drop unknown words	57.33	58.39
Standard + lex-approx	56.13	56.14

Table 4: *Arabic baseline systems*

System	dev1	dev2
Standard phrase-based system	67.43	62.87
+ drop unknown words	67.39	62.83

Table 5: *Turkish baseline systems*

System	dev2	dev3
Standard phrase-based system	67.70	68.60
+ drop unknown words	68.69	69.35
Stardard + MBR	67.03	67.92

Table 6: *French-English baseline systems*

8.3. Domain Adaptation Experiments

As described in section 3, we applied a different formulation of the MAP-based count-smoothing approach we introduced during last year's evaluation. We conducted experiments on both the Arabic-English and French-English tasks using the ISI and Europarl corpora respectively as general purpose models used for backoff when in-domain model probabilities are poorly estimated.

Table 7 compares the IWSLT baseline against the adaptation method we proposed last year and the modification proposed above. In both cases, a gain of $\approx$1 BLEU point can be had. Intuitively, by using relative counts, the new approach allows more refined computation of the λ used to compute the interpolated/adapted probability for each phrase. This method avoids overweighting the gp model when both the $iwslt$ and gp models have relatively few counts.

8.4. Arabic Morphology Experiments

We evaluated the translation results from the CoMMA and CoMMA-MTT processes for both Arabic and Turkish at the aforementioned threshold levels. Tables 8 and 9 show the mean BLEU scores (over ten optimization runs) on the IWSLT-postprocessed truecased output from the Arabic `dev6` and `dev7` data, respectively, by applying the CoMMA and CoMMA-MTT processes. Regardless of the threshold, the CoMMA-MTT process consistently outperformed the standard CoMMA process. Tables 10 and 11 show the mean BLEU scores on the IWSLT-postprocessed truecased output from the Turkish `dev1` and `dev2` data, respectively, by applying the CoMMA and CoMMA-MTT processes. For Turkish, the CoMMA-MTT process outperforms the standard CoMMA process for low thresholds, but it reduces performance for higher thresholds. For a given threshold, the best performing CoMMA and CoMMA-MTT systems from the ten optimization runs were used in system combination experiments in order to choose the final systems to be combined.

CoMMA	Mean BLEU	
Threshold	CoMMA	CoMMA-MTT
0	50.40	51.55
20	53.67	54.44
200	53.88	54.51
2,000	52.44	54.20
10,000	53.06	54.54

Table 8: *Mean BLEU scores for CoMMA and CoMMA-MTT systems versus threshold for the Arabic* `dev6` *data*

CoMMA	Mean BLEU	
Threshold	CoMMA	CoMMA-MTT
0	52.20	52.98
20	53.65	55.10
200	54.82	55.57
2,000	55.02	55.36
10,000	54.96	55.86

Table 9: *Mean BLEU scores for CoMMA and CoMMA-MTT systems versus threshold for the Arabic* `dev7` *data*

CoMMA	Mean BLEU	
Threshold	CoMMA	CoMMA-MTT
0	57.46	59.17
2	59.60	62.61
20	63.87	64.08
200	64.74	63.84
2000	64.56	64.52

Table 10: *Mean BLEU scores for CoMMA and CoMMA-MTT systems versus threshold for the Turkish* `dev1` *data*

CoMMA	Mean BLEU	
Threshold	CoMMA	CoMMA-MTT
0	52.19	54.28
2	55.75	56.00
20	59.10	59.46
200	60.73	59.92
2000	60.20	59.61

Table 11: *Mean BLEU scores for CoMMA and CoMMA-MTT systems versus threshold for the Turkish* `dev2` *data*

8.5. TALK Task Experiments

We ran a number of baseline systems on the talk task data set using using the methods described in prior sections. We used the WMT-supplied segmenters for preprocessing and normalization, and in addition to the IWSLT-supplied data, target-language data from the French Gigaword corpus was used for language modeling in a number of systems. Due to time limitations, we did not evaluate or optimize our system using ASR transcripts as input. In order to perform development experiments, we split the supplied development data into two parts consisting of four talks each (`dev1` = first four, `dev2` = second four). Table 12 summarizes the results of applying to `dev2` .

No single optimization strategy clearly outperforms the other, though the addition of additional language modeling data is a clear benefit ($\approx$0.4-1.0 BLEU). Also, as the supplied talk data is segmented at a breath group/closed-caption level, training continuous ngram language models provides a small performance improvement (lines 5-6 of table 12).

We also ran a set of experiments combining parallel data from the WMT-2010 data set with the supplied talk data and training a combined model. This results in a 1+ point degradation in performance. Due to time limitations we were not able to run comparable experiments using the domain adaptation methods proposed above.

9. Evaluation Summary

As part of this year's evaluation we experimented with improved cross-domain adaptation, improved Arabic morpho-

System	Arabic (dev7)	French (dev3)
IWSLT Model Only (baseline)	55.31	65.51
IWSLT MAP-adapted ([1])	58.85	67.39
IWSLT MAP-adapted (modified)	59.75	68.27

Table 7: *Summary of adaptation experiment results*

System	Optimization Method	dev2
TALK PT + TALK LM	MERT	24.90
TALK PT + TALK LM	MIRA	25.27
TALK PT + TALK LM + Gigaword LM	MERT	25.91
TALK PT + TALK LM + Gigaword LM	MIRA	25.76
TALK PT + Cont. TALK LM + Gigaword LM	MERT	26.15
TALK PT + Cont. TALK LM + Gigaword LM	MIRA	25.87
(TALK + WMT) PT + TALK LM + Gigaword LM	MERT	23.91
(TALK + WMT) PT + TALK LM + Gigaword LM	MIRA	24.43

Table 12: *Summary of TALK task experiments*

logical processing and refinements to our multiple MT combination approach. These developments have helped to improve our system when compared with our 2009 baseline. Our basic system was also applied to the new TALK task.

Table 13 summarizes each of the systems submitted for this year's evaluation and how they compare with our 2009 baselines (when applicable) on the IWSLT09 and TALK test set.

10. Acknowledgments

We would also like to thank Katherine Young for her help in processing the French-Englaish and TALK task data sets and 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., Amizadeh, A.R., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2009 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Tokyo, Japan, 2009.

[2] 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.

[3] 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.

[4] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," In Proc. of MT Summit, 2005.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

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

Arabic-to-English Systems		
System	*Features*	BLEU
AE-primary 2009	2009 baseline	57.17
AE-primary	2010 combined system	58.69
AE-contrast2	2010 best individual system (baseline)	56.58
Turkish-to-English Systems		
System	*Features*	BLEU
TE-primary 2009	2009 baseline	60.01
TE-primary	2010 combined system (without CoMMA)	60.21
TE-contrast1	2010 combined system	60.78
TE-contrast4	2010 best individual system (baseline + MIRA)	58.85
French-to-English Systems		
System	*Features*	BLEU
FE-primary	2010 combined system	63.62
FE-contrast2	2010 best individual system (baseline + MBR)	63.22
TALK Task Systems		
System	*Features*	BLEU
TALK-primary	2010 combined system	26.50
TALK-contrast3	2010 best individual system (baseline + Gigaword)	26.12

Table 13: *Summary of submitted systems*

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki "Online large-margin training for statistical machine translation," In Proc. of EMNLP-CoNLL, 2007.

[22] D. Chiang Y. Marton, and P. Resnik, "Online large-margin training of syntactic and structural translation features," In Proc of EMNLP, 2008.

[23] D. Chiang, K. Knight, W. Wang, "11,001 new features for statistical machine translation," In Proc. NAACL/HLT, 2009.