# The MIT-LL/AFRL IWSLT-2013 MT System[†]

*Michaeel Kazi, Michael Coury,*
*Elizabeth Salesky, Jessica Ray,*
*Wade Shen, Terry Gleason*

MIT/Lincoln Laboratory
Human Language Technology Group
244 Wood Street
Lexington, MA 02420, USA
{michaeel.kazi, michael.coury,
elizabeth.salesky, jessica.ray,
swade, tpg}@ll.mit.edu

*Tim Anderson, Grant Erdmann,*
*Lane Schwartz, Brian Ore, Raymond Slyh,*
*Jeremy Gwinnup, Katherine Young, Michael Hutt*

Air Force Research Laboratory
Human Effectiveness Directorate
2255 H Street
Wright-Patterson AFB, OH 45433
{timothy.anderson.20, grant.erdmann, lane.schwartz,
brian.ore.ctr, raymond.slyh, jeremy.gwinnup.ctr,
katherine.young.1.ctr, michael.hutt.ctr}@us.af.mil

## Abstract

This paper describes the MIT-LL/AFRL statistical MT system and the improvements that were developed during the IWSLT 2013 evaluation campaign [1]. As part of these efforts, we experimented with a number of extensions to the standard phrase-based model that improve performance on the Russian to English, Chinese to English, Arabic to English, and English to French TED-talk translation task. We also applied our existing ASR system to the TED-talk lecture ASR task.

We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2012 system, and experiments we ran during the IWSLT-2013 evaluation. Specifically, we focus on 1) cross-entropy filtering of MT training data, and 2) improved optimization techniques, 3) language modeling, and 4) approximation of out-of-vocabulary words.

## 1. Introduction

During the evaluation campaign for the 2013 International Workshop on Spoken Language Translation (IWSLT-2013) [1] our experimental efforts centered on 1) cross-entropy filtering of MT training data, and 2) improved optimization techniques, 3) language modeling, and 4) approximation of out-of-vocabulary words.

In this paper we describe improvements over our 2012 baseline systems and methods we used to combine outputs from multiple systems. For a more in-depth description of the 2012 baseline system, refer to [3].

The remainder of this paper is structured as follows. Section 2 presents our work on the MT task, and discusses language independent algorithms. Section 3 discusses our MT algorithms for specific language pairs. Section 4 describes final systems and results. Section 5 presents our work on the automatic speech recognition (ASR) task.

## 2. Machine Translation

### 2.1. IWSLT-2013 Data Usage

We submitted systems for the English-to-French, Russian-to-English, Chinese-to-English, Arabic-to-English, and Farsi-to-English MT tasks. We used data supplied by the evaluation for each language pair [2] for training the baseline system, and approved out-of-domain data for the remainder. Unless otherwise noted, we used the optimization data set `dev2010` supplied by IWSLT 2013.

### 2.2. Baseline MT 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].

| | **Training Process** |
|---|---|
| 1. | Segment training corpus |
| 2. | Compute GIZA++, Berkeley and Competitive Linking Alignments (CLA) for segmented data [6] [11] [12] |
| 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 using 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 process*

#### 2.2.1. *Phrase Table Training*

When building our phrase table, we applied Kneser-Ney discounting [5] to the forward and backward translation probabilities of the

[†] 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.

phrases extracted during word alignment. In the past, we have combined multiple word alignment strategies, as described in [6]. For the experiments described here, we used only IBM model 4 or 5 for word alignment (see [7] and [8]), to keep the statistics appropriate for discounting.

### 2.2.2. *Baseline Language Model Training*

During the training process we built n-gram language models (LMs) for use in decoding/rescoring. We performed TrueCasing and repunctuation using 2-gram language models and `disambig` from the SRI toolkit. The MIT Language Modeling Toolkit [10] was used to create interpolated Kneser-Ney LMs in all cases. Additional class-based language models were also trained using `mkcls` [9] for rescoring. Most systems made use of 7-gram language models for rescoring trained on the target side of the parallel text.

### 2.2.3. *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. In our baseline systems, this is done using a standard Powell-like grid search performed on a development set [13].

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 maximum likelihood (ML) or an minimum Bayes risk (MBR) 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 |
| $P(\mathbf{E})$ – 6-gram language model |
| **Rescoring Features** |
| $P_{rescore}(\mathbf{E})$ – 7-gram LM |
| $P_{class}(\mathbf{E})$ – 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*

The `moses` decoder [14] was used for our baseline system. This system serves as the starting point for our all experiments submitted during this year's evaluation. As described in the following sections, we implemented several techniques for generating improved phrase tables and language models, and experimented with using these techniques both individually and in combination.

## 2.3. Cross-Entropy Filtering

Based on the success of cross-entropy training data filtering [15] in last year's evaluation [16], we have continued experimenting with the technique across different language pairs. We used a 3-gram language model based filter, and experimented with LM cross-entropy filtering on each of our systems across different language pairs.

| Corpus | Before Filtering | After Filtering |
|---|---:|---:|
| TED | 141,387 | 141,387 |
| FrEn $10^9$ | 24,116,560 | 824,698 |
| UN | 12,886,831 | 220,066 |
| Europarl | 2,007,723 | 76,554 |
| News Commentary | 137,097 | 1,735 |
| TOTAL | 39,289,598 | 1,264,441 |

Table 3: *Cross-entropy data sizes, using the minimum-perplexity method*

We train a language model on a random subset of the out-of-domain corpus, of the same size as the TED training data. We then sort all sentences in the corpus based on the difference between their cross-entropy given the out-of-domain model and their cross-entropy given the TED language model. The filtered data is taken to be the highest scoring $N$ sentences. We chose the size $N$ in two different ways. First, we simply chose $N$ to be some specific fraction of the data (for example, 5%, 10%, 15%, and 20%). Alternatively, we used an automated approach [17] that uses an information-theoretic estimate of the data size. We train new language models on the best $1/64$, $1/32$, $1/16$, $1/8$, $1/4$, and $1/2$ of the corpus. We selected the filter size that produced the language model with the minimum perplexity on the `dev2010` dataset. To filter the parallel data, we combined the perplexity thresholds that produced the best source and target language models for the `dev2010` dataset.

In general, we aggregated together all out-of-domain parallel data, and performed filtering on the resulting set of sentences; however, we also ran an experiment where automatic filtering was performed independently on each out-of-domain corpus.

In English-to-French, when running automatic filtering on each corpus, this resulted in the selection of 3.2 percent of the overall data for translation model, as shown in Table 3. We also tried manual filtering settings. In all cases, the translation model was fully generated from all of the filtered data.

Our manual filtering results were tested on `tst2010`. In French, we tried 5%, 10%, and 15%. Starting from the system Contrast4 (See Sec 4), changing only the percentage of cross-entropy filtered data, we obtained mean BLEU scores of 31.78, 32.21, and 31.73, respectively. In Russian, we tried 10%, 20%, 30%, and obtained 16.74, 16.64, and 16.62, respectively, compared to a baseline score of 17.13. In Chinese, the same percentages yielded scores of 7.61, 7.37, and 6.57, which were all significantly lower than the baseline average of 10.93. In Arabic, we obtained 24.23, 23.42, and 23.79, with a baseline of 23.90. We stopped increasing the filter size when either performance significantly deteriorated, or our job scheduler terminated Moses (typically when it used more than 200 GB of resident memory).

A list of the corpora can be found in Table 4. In the Chinese and Arabic to English test sets, we used data from the Multi-UN corpora that we sentence aligned using Champollion [18] to obtain the results discussed above. Despite the reasonable sentence pairs produced, we found no significant improvement in the scores.

## 2.4. Improvements to Optimization

We introduce a new optimization technique, "Derivative-free robust error minimization", or DREM. It is distinguished from MERT by its (a) coordinate system, (b) objective function, and (c) other procedural features.

| Corpus | Lang. | Num Sent |
|---|---|---|
| Europarl-v7 | en-fr | 1,495,313 |
| UN | ar-en | 4,743,378 |
| UN | ru-en | 8,344,467 |
| UN | zh-en | 5,948,155 |
| UN | en-fr | 9,018,500 |
| $10^9$ | en-fr | 15,515,787 |
| News Comm. | en-fr | 107,756 |
| Common Crawl | en-fr | 2,563,465 |
| Wiki Headlines | ru-en | 512,000 |

Table 4: A list of the parallel out-of-domain data used in Cross Entropy filtering. Number of sentences is after filtering out sentences of length $> 40$.

| Optimizer | tst2010 | tst2011 |
|---|---|---|
| DREM | 32.82 | 39.35 |
| MERT | 32.41 | - |
| PRO | 32.79 | 39.37 |
| Rampion | 32.88 | 39.10 |

Table 5: Performance of different optimization methods in English-to-French, for the submission system configuration (system details described in Section 4)

With regard to the coordinate system, the weights are tuned on a variance-normalized multi-dimensional unit sphere, rather than in the standard Euclidean space. This incorporates the scale-invariance of the weights and reduces the dimension of the search space by one. Second, it randomizes the coordinate system at every step, which allows it to search in multiple random directions without multiplying the time and effort required.

There are two main novel features of the objective function minimized by DREM. First, the estimated decoder score at a new point considers how far away the new point is from the decode points. A translation that was produced at the closest decode will get full trust, and a translation produced at a more distant decode will be penalized. Second, DREM is not an exhaustive search of the error along a line. Instead, the error function is sampled around the current point and modeled by a (quadratic or linear) function via least-squares regression. This model is minimized around the point, subject to not moving too far away (a "trust-region" constraint). This both reduces metric computations and prevents a sharp valley or spike in the objective function from dominating the behavior, making the result more robust.

Finally, there are two main procedural features of DREM. First, the search for optimal weights is restarted at a few of the most promising of the past decode points, preventing a misstep at an early iteration from having a lingering effect. Second, we have control of the error function minimized. We can manipulate the n-best list into the desired format and use our choice of metrics to define the error function. For this competition, we transformed the n-best list into human-readable text and chose the error function $1 - \frac{1}{2}$(Expected BLEU score $+$ Expected Meteor score).

We compare our results from using DREM on our best systems against MERT, and two other optimization techniques: (a) PRO (Pairwise Ranking Optimization) due to Hopkins and May [19], and (b) Rampion, a technique based on Structured Ramp Loss due to Gimpel and Smith [20]. The results per language can be seen in Tables 5,6,7,8.

| Optimizer | tst2010 | tst2011 | tst2012 | tst2013 |
|---|---|---|---|---|
| DREM | 19.39 | 21.46 | 19.28 | 21.57 |
| MERT | 19.24 | 21.24 | 19.30 | 21.70 |
| PRO | 19.67 | 21.32 | 19.61 | 21.71 |
| Rampion | 18.88 | 20.57 | 18.55 | 20.44 |

Table 6: Performance of different optimization methods in Russian-to-English.

| Optimizer | tst2010 | tst2011 | tst2012 | tst2013 |
|---|---|---|---|---|
| DREM | 11.32 | 15.74 | 13.91 | 14.60 |
| MERT | 11.13 | 14.12 | 12.28 | 13.21 |
| PRO | 11.87 | 15.34 | 13.45 | 14.52 |
| Rampion | 11.10 | 14.19 | 12.32 | 13.22 |

Table 7: Performance of different optimization methods in Chinese-to-English.

### 2.5. Language Modeling

For decoding, a number of different language models were used in various experiments. In general, the procedure was to train a single language model for each domain and subdomain. For example, we would obtain one language model for each news source of the English/French Gigaword corpora. We then either (a) interpolated several language models together, using the MITLM toolkit, or (b) let each language model have its own $\lambda_i$ to be optimized by MERT/DREM/PRO/Rampion, or (c) some combination thereof. Specific submission details can be found Section 4. A list of the monolingual data used can be found in Table 9.

We rescored our n-best lists using both class language models (order-7) and recurrent neural network language models (RNNLM) [21]. The former were trained on the target side of the cross entropy filtered data, while the latter were trained on the monolingual TED data (train.fr/train.en). The recurrent neural network contained 160 hidden units, 300 classes and backpropagation through time of 4. Additionally, some of the Chinese-English systems used a second RNN that contained 10 hidden units and 100 classes. RNN was responsible for substantial gains in most cases. For a summary of its effects, see Table 10.

### 2.6. Lexical approximation

Morphologically rich languages pose a challenge for machine translation systems due to the high number of alternate forms each word may take. Particularly when the size of the training data is small, this creates a sparsity problem for word alignments and results in a higher out-of-vocabulary (OOV) rate. Without specific processing, unknown words are either output without being translated, or are omitted, both of which hurt translation quality. To translate these words, we utilized lexical approximation, which generates alignments for unknown words by approximating those of the closest known word in our GIZA++ word alignments [25].

For each OOV word, we find a series of most-likely candi-

| Optimizer | tst2010 | tst2011 | tst2012 | tst2013 |
|---|---|---|---|---|
| DREM | 25.03 | 25.68 | 27.65 | 26.79 |
| MERT | 24.71 | 25.27 | 27.48 | 26.50 |
| PRO | 24.88 | 25.52 | 27.39 | 26.97 |
| Rampion | 24.05 | 24.83 | 26.53 | 25.83 |

Table 8: Performance of different optimization methods in Arabic-to-English.

| Corpus | English | French |
|---|---|---|
| Europarl-v7 | 55,730,697 | 61,888,789 |
| News Commentary | 3,404,297 | 4,928,120 |
| NewsCrawl '07-'11 | 2,309,306,270 | 616,057,716 |
| FrGigaword v2 | N/A | 827,241,410 |
| EnGigaword v5 | 4,195,862,612 | N/A |
| UN | 361,878,283 | 421,687,471 |
| TED | 2,719,842 | 2,800,512 |
| $10^9$ | 668,269,385 | 810,599,307 |

Table 9: Summary of monolingual training data used.

| Test set | RNN? | French | Russian | Chinese | Arabic |
|---|---|---|---|---|---|
| `tst2010` | N | 32.34 | 19.34 | 11.24 | 25.46 |
|  | Y | 32.82 | 19.39 | 11.32 | 25.49 |
| `tst2011` | N | 38.45 | 21.14 | 15.72 | 26.37 |
|  | Y | 39.35 | 21.46 | 15.74 | 26.23 |
| `tst2012` | N | N/A | 19.33 | 13.92 | 28.41 |
|  | Y | N/A | 19.28 | 13.91 | 28.47 |
| `tst2013` | N | N/A | 21.41 | 14.65 | 28.09 |
|  | Y | N/A | 21.57 | 14.60 | 28.21 |

Table 10: Performance of various systems with and without Recurrent Neural Network language model rescoring. Scores are average BLEU over 10 iterations.

dates from word alignments utilizing character-based Levenshtein distance. We experimented with "approximating" only the in-vocabulary word with the minimum edit distance (1), and the set under a particular threshold (2). In the latter case, we weighted the probabilities of each alignment by the edit distance between the OOV word and its in-vocabulary approximation.

Table 11 shows the reduction in OOV words and the resulting performance improvement by using the above techniques. Due to its higher OOV rate, RU-EN translation benefited more than AR-EN.

|  | Russian |  | Arabic |  |
|---|---|---|---|---|
| Processing | OOV rate | BLEU | OOV rate | BLEU |
| *None* | 2.8% | 21.85 | 2.2% | 24.08 |
| LA (1) | 0.2% | 21.86 | 0.1% | 24.08 |
| LA (2) | 0.0% | 21.98 | 0.1% | 24.11 |

Table 11: OOV Rate and Mean BLEU scores for LA on `tst2013`.

As seen in table 11, LA with a set of values under a threshold performs better than a single replacement, though both provide minor improvement over not processing OOV words. These results are likely due to the fact that while edit distance finds close word forms, there is no guarantee that similar word forms have similar alignments. Further, an OOV word may have no truly similar words in our vocabulary, making its approximation unrelated. In this light, thresholding a set of values provides more possibilities from which a likely alignment to arise.

### 2.7. Development set selection

In past evaluations we have always used the development data given to tune the parameters of our system; however, there is no reason to suspect that tuning performance is independent of the data used, nor that the given TED talks will produce optimal weights for decoding. We try using alternative data for tuning, extracted directly from the TED training data. The sentences not chosen are used for the normal training procedure.

| System | `tst2010` | `tst2011` |
|---|---|---|
| En-Fr | 29.11 | 35.42 |
| En-Fr + Dev | 29.23 | 35.74 |
| En-Fr C4 | 32.01 | 38.75 |
| En-Fr C4 + Dev | 32.01 | 39.22 |
| Ru-En | 18.71 | 21.17 |
| Ru-En + Dev | 18.61 | 20.44 |
| Zh-En | 11.27 | 14.22 |
| Zh-En + Dev | 10.31 | 14.49 |

Table 12: Results with and without dev set selection, using `tst2010` as a target. Scores are average BLEU over 10 iterations, case+punc. C4 refers to the submitted system "Contrastive 4."

Ideally, a development set should resemble the data one expects to decode. Given a language model describing the expected test data, the development set should be drawn from the same distribution. `dev2010` and all evaluation data sets are TED talks, so this is loosely the case already, but we investigate further refinement of the development set. We built a selection algorithm that, given a test set as input, extracts the most similar subset of the TED training data.

Since language models are based off of n-gram counts, our algorithm samples from among the training data to match overall n-gram count frequency. Our algorithm samples to minimize an objective function that loosely resembles the KL-divergence between two language models. (In future work, we will use discounting and explicitly minimize KL divergence.) Let $S$ and $T$ refer to the selection set and input test set, and let $C_S(j)$ indicates the count of n-gram $j$ in the selection set, $C_T(j)$ the analogous count in the test data. The objective function $F(S,T)$ we used is:

$$\begin{aligned} F(S,T) &= \sum_j a\left(\log\frac{C_S(j)}{C_S} - \log\frac{C_T(j)}{C_T}\right) \\ a(x) &= \quad \text{if } x > 0 \text{ then } x \text{ else } -\tfrac{1}{3}x \end{aligned}$$

This pseudo absolute-value $a(x)$ is used to penalize spurious n-grams less than missing n-grams. We tracked n-grams up to order 3, and missing counts in the above formula were given the value $0.1$. Table 12 gives the performance of this algorithm on several experiments.

## 3. MT Language Specific Algorithms

### 3.1. Arabic-to-English Morphological Processing

In our Arabic-to-English MT systems for prior year evaluations [22, 23, 24, 25, 26], we normalized various forms of alef and hamza and removed the tatweel character and some diacritics before applying a light Arabic morphological analysis procedure that we called AP5. Last year, [3] we modified the AP5 procedure to more closely conform to the Arabic Treebank (ATB) segmentation format used in the MADA Arabic morphological analysis, diacritization, and lemmatization system, [27]. This year, we compared the AP5 system to MADA directly, seen in Table 15.

All systems with the rule-based MADA+TOKAN processing outperformed the same system on all test sets with AP5. The degree depended both on the test set and on the optimizer, as seen in Table 15. The most signifant gains were seen using MERT, with a 1.34 BLEU improvement over AP5 on `tst2013`. While both analyses regularize affixes and perform stemming, MADA more pervasively normalizes character variation and segments more heavily than AP5, reducing the OOV rate from 7.0% with AP5 to 2.2% with MADA.

| Segmenter | BLEU |
|---|---|
| charSeg | 10.37 |
| cmuSeg | 9.78 |
| stanSegCTB | 10.72 |
| stanSegPKU | 10.58 |
| charSeg+cmuSeg | 10.71 |
| charSeg+stanSegCTB | 10.83 |
| charSeg+stanSegPKU | 10.66 |

Table 13: Comparison of baseline MT systems for Chinese-English based on vaious word segmenters. The BLEU score is an average over 10 experiments for `tst2010`.

### 3.2. Chinese-to-English Character and Word Segmentation

One challenge of building a machine translation system for Chinese is the absence of spaces between words. We trained systems based on a few different word segmenters for the machine translation task and selected the top performer based on average BLEU score to be our baseline system for this evaluation. The results of our comparison are in Table 13.

The Stanford Chinese Word Segmenter [28] was evaluated using both the Chinese Penn Treebank (CTB) and the Peking University (PKU) segmentation standards. In addition, the CMU LDC Word Segmenter [30] and simply segmenting each individual character were evaluated. GIZA++ was trained using sentences from each segmentation result. Next, the alignment file for each segmenter was further character segmented and combined with the GIZA++ alignments from the character segmenter before being used to create the phrase table.

The Stanford CTB segmenter out-performed the other individual segmenters, and we saw additional gains from combining GIZA++ alignments for this segmenter with the character segmented GIZA++ alignments. As a result, we chose to use the char+stanCTB segmenter for this evaluation.

### 3.3. Russian-to-English Morphological Segmentation

To compensate for the morphological complexity in Russian, we experimented with segmentation. We utilized Morfessor Cat-MAP both to process all the data as well as only for word alignments (WA), [29]. Table 14 shows the mean BLEU scores for individual Russian-to-English MT systems trained on the 2013 training data and tested on the 2010 test set. Morfessor categorizes proposed segments as prefixes, stems, or suffixes. We both kept all generated segments, as well as only stems.

| | RUSSIAN | |
|---|---|---|
| Processing | OOV rate | BLEU |
| *None* | 5.0% | 17.26 |
| Stems for WA only | 3.3% | 16.44 |
| Morfessor Stems | 4.5% | 16.15 |
| Morfessor All Segs | 2.4% | 16.54 |

Table 14: Russian-Specific Experiments, OOV rate and BLEU scores for `tst2010`.

Though processing the data with Morfessor decreased the OOV rate by up to 51.6%, BLEU score decreased. Though word alignments were improved, it was more difficult to organize a greater number of target words into meaningful sentences. Before segmentation, source sentences had on average 14.2 tokens per sentence against 17.12 for English, the relation we would expect given the morphological complexity of Russian. With the best segmentation result (see Table 14) we have 19.3 tokens per Russian sentence. An explanation for poorer performance, then, is that instead of bringing sentence lengths closer together and making fertility closer to 1:1, segmentation widened the gap between the two languages.

## 4. MT Submission Summary

The different experiments we ran in Sections 2 and 3 of this paper played different roles in the submission systems of different languages. In this section we describe the systems that were submitted, and their respective scores. In the tables that follow, the following abbreviations are used:

- **lexDist**: Refers to the moses lexicalized reordering model `wbe-msd-bidirectional-allff`
- **dunk**: Drop unknown words
- **(corpus 1)**$\cdots$**(corpus n) LM**: Linear interpolation of several LMs
- **RNNx**: RNN order $x$
- **FiltLM**: Data for language model filtered via Cross Entropy with TED LM (not interpolated)
- **LA**: Lex approx

System combinations were trained using the `tst2010`. We therefore omit scores for `tst2010` on those systems.

It is also worth noting that systems trained at MITLL used manual cross entropy filter sizes, while those at AFRL used minimum perplexity threshold filter sizes. This is mentioned in the discussion sections.

### 4.1. English-to-French

For French, our best system for `tst2013` (which was submitted as contrastive) used a single order 5 language model from the MITLM toolkit, consisting of the following LMs linearly interpolated (using the MITLM toolkit) on `dev2010`: TED, Europarl-v7, News-Commentary-v7, News-Crawl2007, News-Crawl2008, News-Crawl2009, News-Crawl2010, News-Crawl2011, and the $10^9$ corpus. We found inclusion of LDC French Gigaword v2 did not improve the performance of this language model. The phrase table was filtered at 10% extra data using cross entropy, without using Common Crawl. Our other French systems used a combination of a 6th-order TED language model, and a linearly interpolated language model over LDC Gigaword v2, Europarl, and News Commentary data set. Results are in Table 15. The phrase tables were obtained using cross-entropy filtering with minimum perplexity thresholds on each of the data sets, and including Common Crawl.

### 4.2. Chinese-to-English

Table 15 describes each of the systems we submitted for the Chinese-English portion of the machine translation task. Our primary system is a combination of four different systems.

The best-scoring single system on the `tst2010` data set was the PRO-optimized system, so we decided to set the system combination weights to favor the PRO system over the others. However, the DREM system scored the best for the other data sets. Our contrastive2 submission had a significantly higher weight for the PRO system compared to the weight for our primary submission. Perhaps we would have seen even higher scores for the `tst2013` data set if we had set the weights higher for the DREM system. When performing system combination, the primary and contrastive2 systems used different prior weights during training.

### 4.3. Russian-to-English

Table 15 describes each of the systems we submitted for the Russian-English portion of the machine translation task. Our primary system is a combination of three different systems. Our best system on `tst2013`, improperly tokenized when submitted as Contrast3$^{\dagger}$, used a single 4th-order language model from the MITLM toolkit, consisting of the following LMs linearly interpolated (using the MITLM toolkit) on `dev2010`: TED, MultiUN, Wikipedia headlines, and LDC English Gigaword v5. The phrase table kept 20% of extra data using cross entropy filtering, and used Wikipedia Headlines + United Nations data. Our other Russian systems used a combination of a 6th-order TED language model, and a linearly interpolated language model over the LDC Gigaword, MultiUN, and News-Crawl2007, News-Crawl2008, News-Crawl2009, News-Crawl2010, News-Crawl2011 corpora. For the cross entropy filtering, we used News-Commentary-v7 and the News-Crawl corpora with minimal perplexity thresholds.

### 4.4. Arabic-to-English

Before the deadline, we were only able to submit results for AP5 with various optimization. However, we include in the table results for MADA, which significantly outperforms the submitted systems.

## 5. Automatic Speech Recognition

Acoustic training data for our ASR systems were harvested from 838 TED Talks. We applied the same alignment and closed caption filtering process as IWLST 2011 [26], except that each utterance was padded by a maximum of 0.25 seconds (instead of 2.0 seconds) and the filtering threshold was set to 30% WER (Word Error Rate). This yielded 166 hours of audio.

A GMM-HMM system was trained using Perceptual Linear Prediction (PLP) features. This system was developed using the same training procedure as our IWSLT 2011 system, except that this year we applied mean and variance feature normalization on a per speaker basis. The updated data partition and feature normalization yielded a 1.0% WER reduction on `dev2010`, `tst2010`, and `dev2012`.

A secondary GMM-HMM (GMM-HMM-2) system was trained in a similar fashion as the prior, but using the CMU Pronouncing Dictionary [31]. Missing dictionary entries from the training data were generated by training a grapheme-to-phoneme model using Sequitur G2P, an open-source grapheme-to-phoneme converter [32]. This system did not quite reach the performance of the other GMM-HMM system, however, use of the stress markings included in the CMU Pronouncing Dictionary are being furthered explored to evaluate their impact on performance, and initial tests show a decrease in WER on `dev2010` of approximately 0.3% as compared to ignoring the stress markings.

A hybrid Deep Neural Network (DNN)-HMM speech recognition system was developed using Theano [33] and a version of HTK that we modified according to the method of [34]. The DNN included 5 hidden layers, each of which had 1000 neurons with logistic activation functions. A context window of 9 frames was used at the input, and the output included 6000 units corresponding to the shared states of our GMM-HMM system. The feature set consisted of 13 PLPs with delta and acceleration coeffcients, and all features were normalized to zero mean and unit variance on a per speaker basis. Training was performed using layer growing back propagation [35] with a minibatch size of 512, and an initial learning rate of 0.008 that was halved after each epoch once the improvement in accuracy on the cross validation partition fell below 0.5%. A second DNN was trained on PLP features that were transformed using Constrained Maximum Likelihood Linear Regression (CMLLR). This system applied a single transform per speaker.

LM data selection was implemented using the same procedure as our IWSLT 2012 system [3]. Interpolated trigram and 4-gram LMs were estimated on TED, 1/8 of Gigaword, and 1/4 of News 2007–2012 using the SRILM Toolkit.[1] Compared to a trigram LM trained on all of the available data, applying data selection reduced the WER of our GMM-HMM system by 0.4% on `dev2010`, `tst2010`, and `dev2012`. Recurrent Neural Network Maximum Entropy (RNNME) LMs were trained on 1/16 of Gigaword and 1/8 of News 2007–2012 using the RNNLM Toolkit [21]. Each network included 160 hidden units, 300 classes in the output layer, 4-gram features for the direct connections, and a hash size of $10^9$. The LM vocabulary included 95000 words.

A neural network based Speech Activity Detector (SAD) was developed using Theano. The SAD was trained on 22 hours of TED data and 5 hours of public domain music downloaded from Wikimedia Commons,[2] the United States Air Force Band,[3] and the Open Goldberg Variations project.[4] The network included a context window of 21 frames on the input, 1 hidden layer of 500 neurons with logistic activation functions, and 3 output units corresponding to speech, silence/noise, and music. The feature set consisted of 13 PLPs with delta and acceleration coeffcients, and all features were globally normalized to zero mean and unit variance. Training was perfomed using the same procedure as the DNNs.

Automatic segmentation of the test data was performed by evaluating the SAD, applying a dynamic programming algorithm to choose the best sequence of states, and padding the speech end points by 0.15 seconds. The speech segments from each talk were clustered using the MIT-LL GMM-based speaker recognition software package. Compared to the manual segmentation provided in the reference files, automatically segmenting the test data increased the WER of our GMM-HMM system by 0.7% on `dev2010`, `tst2010`, and `dev2010`.

Initial transcripts of the test data were produced using the hybrid DNN-HMM system. Next, CMLLR transforms were estimated for the GMM-HMM system and the second hybrid DNN-HMM system. Recognition lattices were produced for each system and then rescored with the interpolated 4-gram LM. The final transcripts were produced by rescoring n-best lists with the RNNME LMs.

System combination was performed as in IWSLT 2012 [3] using a Confusion Network Combination system (CNC). Confusion networks for combined systems are generated from rescored n-best lists of size 1000. The confusion networks are aligned with each other, and this alignment is used to merge the individual system's confusion networks into one. Each system is weighted (weights generated from a Powell-like grid search) and acoustic model and language model scores of each are combined. Due to time constraints, tests for our system combination were performed on `dev2010` using the GMM-HMM and DNN-HMM, but results were not submitted. Table 16 shows WERs for the individual systems and WER for the combined system using this methodology.

Table 17 shows results on `dev2010` for the DNN-HMM system, the GMM-HMM system, and the GMM-HMM-2 system before and after RNNLM rescoring. Table 18 shows the progress of our current systems against our best submission from IWSLT 2012 [3] on `tst2011` and `tst2012`. Results for `tst2013` on our current submissions are also shown.

[1] Available at: http://www.speech.sri.com/projects/srilm

[2] Available at: http://commons.wikimedia.org

[3] Available at: http://www.usafband.af.mil

[4] Available at: http://www.opengoldbergvariations.org

| System | Description | tst2010 | tst2011 | tst2012 | tst2013 |
|---|---|---|---|---|---|
| | English-to-French | | | | |
| primary | DREM + UNTen9EuroNCommCC FiltLM + Giga LM + RNN3 + dunk | 32.82 | 39.35 | 39.76 | 37.05 |
| contrast1 | Ramp + UNTen9EuroNCommCC FiltLM + Giga LM + RNN3 + dunk | 32.88 | 39.10 | 39.94 | 37.12 |
| contrast2 | Primary + Contrast4 | N/A | 38.97 | 39.70 | 37.32 |
| contrast3 | PRO + UNTen9EuroNCommCC FiltLM + Giga LM + RNN3 + dunk | 32.79 | 39.37 | 39.70 | 37.41 |
| contrast4 | MERT + tedUNTen9NewsCrawlEuro LM + dunk | 32.21 | 38.90 | 39.83 | 37.58 |
| contrast5 | Primary + Contrast3 | N/A | 39.27 | 39.97 | 37.21 |
| | Chinese-to-English | | | | |
| primary | contrast5 + contrast3 + contrast6 + contrast4 | 11.46 | 15.92 | 14.05 | 14.85 |
| contrast1 | contrast5 + contrast3 + contrast4 | 11.40 | 15.90 | 13.59 | 14.77 |
| contrast2 | contrast5 + contrast3 + contrast6 + contrast4 | 11.53 | 16.00 | 14.00 | 14.77 |
| contrast3 | PRO + tedGiga LM + RNN3 + lexDist + dropunk | 12.03 | 15.14 | 13.50 | 14.36 |
| contrast4 | MERT + tedUNGiga LM + RNN5 | 11.72 | 14.64 | 12.35 | 13.25 |
| contrast5 | DREM + tedLM + Giga LM + RNN3 + lexDist + dunk | 11.47 | 15.85 | 13.93 | 14.61 |
| contrast6 | MERT + tedUNGigaEuroNComm LM + RNN5 + dunk | 11.20 | 14.87 | 12.63 | 13.50 |
| | Russian-to-English | | | | |
| primary | contrast1 + contrast2 + contrast3 | N/A | 21.49 | 19.61 | 21.65 |
| contrast1 | PRO + tedNewsCrawlCC FiltLM + Giga LM + RNN3 + LA2 + dunk | 19.67 | 21.32 | 19.61 | 21.71 |
| contrast2 | MERT + tedNewsCrawlCC FiltLM + Giga LM + RNN3 + LA2 + dunk | 19.24 | 21.24 | 19.30 | 21.70 |
| contrast3† | MERT + tedUNWiki LM + LA2 + dunk | 19.42 | 21.68 | 19.58 | 22.13 |
| contrast4 | DREM + tedNewsCrawlCC FiltLM + Giga LM + RNN3 + LA2 + dunk | 19.39 | 21.46 | 19.28 | 21.57 |
| | Arabic-to-English | | | | |
| primary | DREM + lexDist + ted LM + giga LM + RNN3 + AP5 + dunk | 25.03 | 25.66 | 27.66 | 26.64 |
| contrast1 | PRO + lexDist + ted LM + giga LM + RNN3 + AP5 + dunk | 24.88 | 25.81 | 27.52 | 27.27 |
| contrast2 | MERT + lexDist + ted LM + giga LM + RNN3 + AP5 + dunk | 24.71 | 24.95 | 27.27 | 26.22 |
| contrast3 | MERT + lexDist +tedUNGigaEuro LM + AP5 + dropunk | 24.36 | 24.96 | 26.95 | 25.77 |
| MADA | DREM + lexDist + ted LM + giga LM + RNN3 + MADA + dunk | 25.49 | 26.23 | 28.47 | 28.21 |
| MADA1 | PRO + lexDist + ted LM + giga LM + RNN3 + MADA + dunk | 25.21 | 26.41 | 27.92 | 27.70 |
| MADA2 | MERT + lexDist + ted LM + giga LM + RNN3 + MADA + dunk | 25.14 | 26.01 | 27.97 | 27.56 |

Table 15: All Submission Systems. †For this system, fixed tokenization issue after submission.

| dev2010 | | |
|---|---|---|
| DNN-HMM | GMM-HMM | Combined |
| 13.9 | 14.5 | 13.7 |

Table 16: WER for individual DNN-HMM and GMM-HMM systems and their system combination on dev2010 (automatic segmentations, without RNNLM rescoring).

| | dev2010 | |
|---|---|---|
| | without RNNLM | with RNNLM |
| DNN-HMM | 14.1 | 12.9 |
| GMM-HMM | 13.8 | 12.8 |
| GMM-HMM-2 | 17.2 | 15.6 |

Table 17: WER without and with RNNLM rescoring on dev2010 (manual segmentations).

| | tst2011 | tst2012 | tst2013 |
|---|---|---|---|
| *IWSLT 2013* | | | |
| DNN-HMM | 10.6 | 11.3 | 15.9 |
| GMM-HMM | 9.7 | 11.0 | 16.7 |
| GMM-HMM-2 | 12.5 | 13.9 | 23.0 |
| *IWSLT 2012* | | | |
| GMM-HMM | 12.6 | 14.3 | N/A |

Table 18: WER of IWSLT 2013 submissions on tst2011, tst2012 versus our best 2012 system. tst2013 is also shown.

## 6. Acknowledgments

We would like to thank members of the Wright-Patterson AFB, and of the Human Language Technology group at MIT Lincoln Lab for their support and state of the art computer systems.

## 7. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, S. Stüker "Overview of the IWSLT 2012 Evaluation Campaign," In *Proc. of IWSLT*, Hong Kong, HK, 2012.

[2] M. Cettolo, C. Girardi, and M. Federico "WIT3: Web Inventory of Transcribed and Translated Talks," In *Proc. of EAMT*, pp. 261-268, Trento, Italy, 2012.

[3] J. Drexler, W. Shen, T. Gleason, T. Anderson, R. Slyh, B. Ore, and E. Hansen, "The MIT-LL/AFRL IWSLT-2012 MT System," in *Proceedings of IWSLT 2012,* (Hong Kong, HK), 2012.

[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] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proceedings of EMNLP 2006,* (Sydney, Australia), July 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] 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.

[8] 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.

[9] Och, F. J. “An Efficient Method for Determining Bilingual Word Classes,” Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics; EACL’99, pp. 71-76. Bergen, Norway, June 1999.

[10] Bo-June (Paul) Hsu and James Glass, “Iterative Language Model Estimation: Efficient Data Structure and Algorithms,” In Proc. Interspeech, 2008.

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

[12] 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.

[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.

[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] S. Mansour *et al.*, “Combining Translation and Language Model Scoring for Domain-Specific Data Filtering,” in *Proc. International Workshop on Spoken Language Translation*, San Francisco, USA, 2011.

[16] E. Hasler *et al.*, “The UEDIN Systems for the IWSLT 2012 Evaluation,” in *Proceedings of IWSLT 2012,* (Hong Kong, HK), 2012.

[17] R. C. Moore and W. Lewis, “Intelligent Selection of Language Model Training Data,” in *Proceedings of the ACL 2010 Conference Short Papers,* (Uppsala, Sweden), 2010.

[18] X. Ma, “Champollion: A Robust Parallel Text Sentence Aligner,” in *Proceedings of LREC 2006,* (Genova, Italy), 2006.

[19] M. Hopkins and J. May, “Tuning as ranking,” in *Proc. of EMNLP 2011* (Edinburgh, Scotland, UK), July 2011.

[20] K. Gimpel and N. A. Smith, “Structured Ramp Loss Minimization for Machine Translation,” in *Proceedings of NAACL 2012,* (Montreal, Canada) June 2012.

[21] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, “Strategies for Training Large Scale Neural Network Language Models,” in *Proc. Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, 2011.

[22] 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.

[23] 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.

[24] Shen, W., Delaney, B., Aminzadeh, 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.

[25] Shen, Anderson, T., Slyh, R., and Aminzadeh, A.R., “The MIT-LL/AFRL IWSLT-2010 MT System,” In Proc. Of the International Workshop on Spoken Language Translation, Paris, France, 2010.

[26] A. R. Aminzadeh, T. Anderson, R. Slyh, B. Ore, E. Hansen, W. Shen, J. Drexler, and T. Gleason, “The MIT-LL/AFRL IWSLT-2011 MT system,” in *Proceedings of IWSLT 2011,* (San Francisco CA), December 2011.

[27] R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin, “Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking,” in *Proceedings of ACL-08: HLT, Short Papers,* (Columbus OH), June 2008.

[28] P. C. Chang, M. Galley, and C. D. Manning, “Optimizing Chinese Word Segmentation for Machine Translation Performance,” in *Proceedings of the Third Workshop on Statistical Machine Translation,* (Columbus, OH), June 2008.

[29] M. Creutz, K. Lagus. “Unsupervised models for morpheme segmentation and morphology learning,” in *ACM Transactions on Speech and Language Processing*, 4(1):1-34, 2007.

[30] Z. Wu. “LDC Chinese Segmenter,” *http://www.ldc.upenn.edu/Projects/Chinese/segmenter/mansegment.perl*, 1999.

[31] R. Weide., “The CMU pronouncing dictionary,” *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*, 1998.

[32] M. Bisani., “Sequitur G2P: A trainable Grapheme-to-Phoneme converter,” *http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html*, 2011.

[33] J. Bergstra *et al.*, “Theano: A CPU and GPU Math Expression Compiler,” in *Proc. Python Scientific Computing Conference (SciPy)*, Austin, TX, 2010.

[34] G. Dahl, D. Yu, L. Deng, and A. Acero, “Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, January, 2012.

[35] F. Seide, G. Li, and D. Yu, “Conversational Speech Transcription Using Context-Dependent Deep Neural Networks,” in *Proc. Interspeech*, Florence, Italy, 2011.