The MITLL-AFRL IWSLT 2015 Systems[†]

Michaeel Kazi[1], *Brian Thompson*[1], *Elizabeth Salesky*[1], *Timothy Anderson*[2], *Grant Erdmann*[2], *Eric Hansen*[2], *Brian Ore*[2], *Katherine Young*[2], *Jeremy Gwinnup*[2], *Michael Hutt*[2], *Christina May*[2]

[1]MIT Lincoln Laboratory
Human Language Technology Group
244 Wood Street
Lexington, MA 0220, USA
{first.last}@ll.mit.edu

[2]Air Force Research Laboratory
Human Effectiveness Directorate
2255 H Street
Wright-Patterson AFB, OH 45433
{first.last.*}@us.af.mil

Abstract

This report summarizes the MITLL-AFRL MT, ASR and SLT systems and the experiments run using them during the 2015 IWSLT evaluation campaign. We build on the progress made last year, and additionally experimented with neural MT, unknown word processing, and system combination. We applied these techniques to translating Chinese to English and English to Chinese. ASR systems are also improved by refining improvements developed last year. Finally, we combine our ASR and MT systems to produce a English to Chinese SLT system.

1. Introduction

During the evaluation campaign for the 2015 International Workshop on Spoken Language Translation (IWSLT '15) [1] our experimental efforts in machine translation (MT) centered on 1) the addition of hierarchical decoding systems 2) reranking n-best lists with a neural net encoder-decoder 3) post-processing of unknown words in translation output and 4) system combination.

Experimental efforts for the automatic speech recognition (ASR) task focused on using cutting edge neural net techniques and the combination of HTK and Kaldi-based ASR systems.

We combine both efforts to produce a system for the spoken language translation (SLT) task. Various segmentation and punctuation strategies were explored.

This paper is structured as follows. Section 2 presents our work on the MT task, and discusses each of the techniques mentioned above, ending with a discussion of submitted systems. Our work on the ASR task is discussed in Section 3. Finally, our work on the SLT task is discussed in Section 4.

[†]This material is based upon work supported by the Air Force Research Laboratory under Air Force Contract No. (FA8721-05-C-0002 and/or FA8702-15-D-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Air Force Research Laboratory.

2. Machine Translation

2.1. Data Usage

Unless otherwise noted, data described in this section originates from the WMT15 website[1]. We used the parallel in-domain data supplied by WIT3 [2]. In Chinese–English, we additionally used the Yandex corpus[2], Common Crawl, Wiki Headlines, News Crawl, and the LDC Gigaword corpus as sources of monolingual English data for language model training. In English–Chinese we utilized the Chinese portion of the MultiUN corpus as an additional source of language model training data.

2.2. Data Preprocessing and Cleanup

As in past years, we applied a cleaning process to the training data as previously described in [3]. Chinese was segmented with the Stanford Segmenter [4] using both Chinese Treebank (CTB) and Peking University (PKU) models.

2.3. Training

2.3.1. Phrase and Rule Table Training

We used the default Moses scripts when training phrase and rule tables. For Chinese to English, we increased the size of the training corpus by concatenating output from both the CTB and PKU segmentation models while simply repeating the English portion of the corpus. This allows us to extract phrases for a greater number of phrases than one segmentation alone. We also experimented with outputting the k-best segmentation choices for a model while repeating the English portion.

Phrase and rule tables are trained with default Moses scripts or our custom MT pipeline driver. Good-Turing smoothing[5] was applied to both rule and phrase tables.

[1]http://www.statmt.org/wmt15/translation-task.html
[2]https://translate.yandex.ru/corpus?lang=en

2.3.2. Language Model Training

We reuse our BigLM15 from our WMT15 shared translation task submission[6] as our main English machine translation language model. The English data sources listed in Section 2.1 were used to train a very large 6-gram language model. For Chinese, we take a similar approach to English, using the TED in-domain parallel training data and the Chinese portion of the MultiUN corpus. `kenlm` [7] was used to train 6-gram models in both languages. These models were then binarized and stored on local solid-state disks for each machine in our cluster to improve load time and reduce fileserver traffic.

2.4. Baseline MT System

Our system implements a fairly standard statistical machine translation (SMT) architecture. It consists of the following:

- Moses phrase-based [8] or hierarchical decoding with the incremental-search algorithm [9]
- Stanford Chinese character segmentation [4]
- Hierarchical **mslr** lexical reordering [10] for phrase-based systems
- Minimal phrase table [11]
- 7-gram brown-cluster language model with 80 classes
- BigLM15 [6] for English, consisting of WMT newscrawl data, europarl, news commentary
- Drem optimization [12]
- Recurrent neural net language model (RNNLM) rescoring [13]

2.5. Neural MT methods

2.5.1. Chinese to English

We reranked our n-best lists using an end-to-end neural MT system: our own in-house Torch7 [14] implementation of Sutskever et al [15]'s LSTM encoder-decoder approach. This system was trained using varying amounts of out-of-domain UN data, followed by training on TED data. In the following table, the UN sentences were ranked according to bilingual cross-entropy difference [16] (using RNNLM for the language model component) and the top N were chosen to pretrain the network. Once validation error settled down, the networks were then trained over the 200,000 TED training examples. The different models that have been trained can be seen in Table 1. In common among all of them were vocab selection: vocab entries were taken if they appeared at least 10 times in TED, or 100 times in UN, or 5 times in TED and 20 times in UN. Reranking results for individual systems, as well as pairwise combinations, can be seen in Tables 1 and 2.5.1.

Using the best scoring encoder-decoder (#3), and the best scoring combination, we were able to rerank the n-best list from our best hierarchical moses system, and achieved significant gains. In particular, we increased the score from 16.94 to 17.60 cased BLEU on tst2013 for our best hierarchical system (see Table 4).

id	d	N	dev ppl	BLEU
0	4	2.5M	27.15	16.89
1	1	1M	30.01	16.68
2	2	5M	25.54	16.92
3	2	1M	28.26	16.97
4	1	2.5M	27.98	16.62
5	2	2.5M	24.21	16.73

Table 1: Perplexity on `dev2010` (network validation cost), and cased BLEU on `tst2013`. d = LSTM depth and N = cross-entropy filter size for UN data.

	0	2	3	4	5
0	-	16.90	16.89	16.77	16.93
2		-	17.00	16.69	16.96
3			-	16.75	16.89
4				-	16.84
5					-

Table 2: Cased BLEU on `tst2013` for combinations of encoder-decoders, numbered as in Table 1.

2.5.2. English to Chinese

For the English to Chinese task, we achieved gains by using the Neural Network Joint Model (NNJM [17]), and additionally by reranking using RNNsearch [18], the Montreal LISA-lab attention model system. The NNJM was trained using our own in-house implemention in Theano [19]. We integrated NNJM decoding into Moses as a feature function, utilizing self-normalization and precomputation to allow reasonable runtimes. This gave us a gain of approximately 0.2 BLEU over a strong baseline including factored models and RNN rescoring. For RNNsearch, we used GroundHog[3] to train a model, and to compute scores over an n-best list produced by our conventional MT systems. The network sizes used were GroundHog defaults. Like our Chinese to English Torch system, the GroundHog system used MultiUN data in the same way. This gave us an additional 0.4 BLEU gain. A summary of results can be seen in Table 3.

2.6. System combination

This year we experimented with system combination techniques based on Rosti et al [20], a well established technique in machine translation. Our only additional

[3] `https://github.com/lisa-groundhog/GroundHog`

En-Zh System	char BLEU
Baseline	20.37
+ 400 Class Factored LM	20.52
+ RNNLM	21.23
+ NNJM	21.42
+ GroundHog 2M	21.64
-/+ GroundHog 4M	21.85

Table 3: English–Chinese system additions

id	Description	BLEU
0	Hiero, 6-iter Drem Dev10, CLM	16.50
1	(0) + fixed wide beam	16.88
2	(0) + bigdev	16.94
3	(2) + enc-dec	17.60
4	(2) + 3-iter Drem variation	16.60
5	(2) + 6-iter Drem variation	16.66
6	Hiero Incsearch, bigdev, no rescoring	15.58
7	PB, bigdev, no rescoring	14.38
8	PB, ted+nyt CLMs, enc-dec	17.06

Table 4: Some notes on the systems: CLM = brown cluster language model, bigdev = `dev2010 + tst2010 + tst2011 + tst2012`. TED factored LM has 80 classes, nyt (LDC English Gigaword) has 600. All used a variation of LMs trained on WMT'15 data.

contribution was in sub-selecting systems with which to perform system combination. Among our different collaborators, we managed to produce over 30 systems with 400+ decode outputs. With the goal of choosing only 9, we first filtered out systems with scores less than some minimum acceptable value (in our case, 16.50 cased BLEU on `tst2013`). Then, we constructed a distance metric as $1-\text{BLEU}(x, y)$ and performed k-medoids clustering to choose systems that were sufficiently different from each other.

Table 4 lists different systems used for combination, and Table 5 lists a sampling of combinations tried and their case-sensitive BLEU scores on `tst2013`.

Combo id	Systems Used	tst2013 BLEU
0	0+1+2+4+8	17.62
1	0+1+3+5+8	17.64
2	0+1+2+4+5+8	17.64
3	0+1+3+4	17.66
4	0+1+5+8	17.74

Table 5: Top 5 systems out of system combination

2.7. Unknown Word Processing

As in our WMT15 submission [6], we employed unknown word post-processing to handle any unknown words in the translation instead of simply dropping these words. To test the effectiveness of this approach, we decode all test sets where references are available with a bare-bones Moses hierarchical decoding system where no rescoring features are employed. The resulting gains measured in uncased BLEU are shown in Table 6. We note that the improvements in BLEU are smaller for the Chinese–English language pair when compared to our efforts in processing unknown words in other language pairs, such as Russian–English[6], but we feel that employing these processes are still worthwhile due to the positive impact on readability of the machine translation output. Our technique adapted to Chinese–English is described in the following section.

2.7.1. Chinese to English post-processing

The named entity list used for named entity post-processing comes from manual translations of named entities found in train 2014. It was expanded by adding versions of the Chinese name with the common nouns stripped off. A list of 29 typical common nouns endings of named entity phrases was compiled. Common nouns like: 病 (disease), 县 (county), 族 (race/people), 实验室 (laboratory), 湖 (lake), 集团 (corporation), 群岛 (archipelago) can sometimes be optionally included or omitted by the speaker or optionally split off of entities by word segmenters or named entity taggers.

The output is searched for words containing any Chinese characters. Any unknown word consisting of a single character is deleted since single-character entities are rare in this domain (and segmentation errors are a more common explanation for unknown single-character words). If the word is not found in the named entity word list, the list is searched again for the entity with common nouns stripped. Remaining unknown words are deleted from the output.

Test Set	base. BLEU	post. BLEU	Δ BLEU
`tst2013`	16.09	16.19	+0.10
`tst2012`	13.64	13.65	+0.01
`tst2011`	15.21	15.29	+0.08
`tst2010`	12.43	12.50	+0.07

Table 6: NE post-processing improvement measured in uncased BLEU.

2.8. Submission

Our primary Chinese–English MT submission is system #3 in Table 4. We submitted system #4 in Table 5 as contrastive. For English–Chinese, the primary system

was the last entry in Table 3.

These systems were used to decode the `tst2014` and `tst2015` test sets. Results from scoring performed by the workshop organizers are listed in Table 7 including baseline system scores as determined by the organizers.

System	Lang Pair	Test Set	BLEU
Baseline	Zh-En	tst2014	11.43
Primary	Zh-En	tst2014	14.13
Contrastive	Zh-En	tst2014	13.35
Baseline	Zh-En	tst2015	13.59
Primary	Zh-En	tst2015	16.86
Contrastive	Zh-En	tst2015	15.05
Baseline	En-Zh	tst2014	17.74
Primary	En-Zh	tst2014	18.51
Baseline	En-Zh	tst2015	21.86
Primary	En-Zh	tst2015	24.31

Table 7: Official results measured in cased BLEU.

3. ASR

Acoustic training data for our ASR sytems were harvested from 1787 TED talks. We applied the same alignment and closed caption filtering process as we did in IWSLT 2013 [21], yielding 336 hours of audio.

An i-vector system was first developed on the TED data using Hidden Markov Model Toolkit(HTK)[4] Mel-Frequency Cepstral Coeffiect (MFCC) features and the MIT-LL i-vector software. The elements of the 50 dimensional MFCC vector were based on those used by the ALIZE toolkit [22]. Non-speech frames were removed using the word alignments from the closed caption filtering process, and the features were normalized to zero mean and unit variance on a per-speaker basis. The universal background model included 1024 Gaussians with diagonal covariances, and the i-vector dimension was set to 100. Lastly, the Eigen Factor Radial method [22] was applied to normalize the i-vectors.

A hybrid deep neural-net(DNN) - hidden Markov model(HMM) speech recognition system was developed using Theano and a version of HTK that we modified according to the method of [23]. A context window of 9 frames was used on the input, and the speaker-specific i-vector was appended to each set of stacked features [24]. The feature set consisted of 24 log filterbank outputs with delta and acceleration coefficients; the features were normalized to zero mean and unit variance on a per-speaker basis. The DNN included 5 hidden layers with 1024 rectified linear units per hidden layer and 8000 output units. The network weights were initialized as suggested in [25]. Cross-entropy training was performed using a minibatch size of 512 and an initial learning rate of 0.0005 that was adjusted according to the QuickNet newbob algorithm.[5]

LM data selection was implemented using the same procedure as our IWSLT 2014 system. Interpolated trigram and 4-gram LMs were estimated on TED, 1/8 of Gigaword, and 1/8 of News 2007–2014 using the SRILM Toolkit. A RNN maximum entropy LM was estimated on the same set of training texts using the RNNLM Toolkit. The 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 100,000 words.

Automatic segmentation of the test data was performed using the same procedure as in IWSLT 2014[3], except that we padded the speech end points by 0.25 seconds (instead of 0.15 seconds). Recognition lattices were produced using HDecode with the trigram LM and then rescored with the 4-gram LM. Next, 1000-best lists were extracted from each lattice and rescored with the RNN LM. The final LM scores were obtained by linearly interpolating the log probabilites from the 4-gram and RNN LM. Interpolation weights of 0.25 for the 4-gram and 0.75 for the RNN were chosen based on results from previous experiments.

Adaptation data was selected for each speaker using confidence scores [26]. In our work, we estimated confidence scores at the acoustic frame level by aligning the 20-best hypotheses for each utterance and counting the number of matching HMM shared states. Next, speaker-dependent DNNs were estimated on frames with a confidence score of 0.9 or higher. For each speaker, the initial DNN was updated using a learning rate of 0.0000625 and a single epoch of training. The test set was then decoded a second time and LM rescoring was reapplied.

A second ASR system was built using the Kaldi open source speech recognition toolkit [27]. This system was based on the LIUM recipe as released with Kaldi under egs/tedlium/s5. The details of the particular system used for the IWSLT 2015 Kaldi-based ASR system are as follows. The acoustic model training data and LM data matched exactly what was used as previously described in the HTK ASR system. The first step was to build a network to produce bottleneck (BN) features [28]. MFCCs from 40 filterbanks and 3 pitch features were used as input to a neural network of 2 hidden layers each of dimension 1500 with a 40 dimension BN layer producing the output features. These 40 BN features were then used to build a GMM-HMM. Speaker adaptive training was then conducted on this GMM-HMM using feature-space maximum likelihood linear regression (fMLLR) transforms. These models were then used to train a DNN of the Deep Belief Network (DBN)

[4] http://htk.eng.cam.ac.uk

[5] http://www.icsi.berkeley.edu/Speech/faq/nn-train.html

ASR System	Decode	4-gram	4-gram+RNN
HTK first-pass	13.7	13.0	11.9
HTK	11.3	10.9	10.0
Kaldi	13.3	12.6	11.4

Table 8: English `tst2013` WER.

variety described as having 6 hidden layers with 2048 neurons per layer. Four additional iterations using the state-level Minimum Bayes Risk (sMBR) discriminative crierion were then executed. This system was evaluated using the trigram LM to produce recognition lattices, which were then rescored with the 4-gram and RNN LMs as described for the HTK system.

Table 8 shows the WER of each system on `tst2013` after evaluating the decoder, rescoring with the 4-gram LM, and interpolating the 4-gram and RNN LM scores. For comparison purposes, we included the results of the HTK system prior to updating the weights of the DNN (denoted as HTK first-pass). The final hypothesis was selected by applying N-best ROVER to the output from the HTK system and the Kaldi system. This yielded a 9.4% WER on `tst2013` and a 6.6% WER on `tst2015`.

4. SLT

New for this year, we combine our efforts in ASR and MT to produce an entry to the SLT task for the English-Chinese language pair. We use the rover output from system combination from the ASR task and translate it with a variant of our best English-Chinese MT system. For segmenting the output, we used the segmentations produced by the ASR system, based on lengths of pauses. To repunctuate the ASR output, we created a classifier based on a recurrent neural network. For each word, the classifier reports which punctuation, if any, follows it. The output layer is a softmax over a limit set (period, comma, question mark, exclamation point, and no punctuation). The inputs to the classifier are a gated recurrent unit [29] hidden state for the word in question, as well as its word vector and the word vectors for the following two words after. Our experimentation was quite limited, but we observed that (a) having the word vector as well as the recurrent state for the current word was helpful, and (b) three layer deep gated recurrent unit worked best out of 1-4. The system was trained on the English side of TED data, with 600-dimensional word vector size, and vocabulary of about 60K words. We did not try any other repunctuation techniques.

One of our alternate approaches for adapting ASR output to MT involves taking the output of the ASR system when decoding `dev2010` then using the `mwerAlign`[30] program to then fit the ASR output segments to the English portion of the `dev2010` tuning set. We then tune the MT system with this new dev set in order to better match the ASR English output to the English–Chinese MT system. We submitted this as a contrastive system.

5. Future Research

For future research, we are beginning to look at the metadata for the individual TED talks. We have looked at the distribution of dev, test and training talks by date posted, for example. We are particularly interested in the way different translators may affect the quality of the translations. The TED website assigns a translator ID to each translator, which can be used to isolate his or her talks. The IWSLT files provide the translator metadata for the training files; for the dev and test files, it was necessary to look up the translator annotations in the source files on the TED Talks website.

We looked at the output for each talk in the test files individually, and compared scores for different translators. For example, the scores in cased BLEU for `tst2011` are shown in Tables 9 and 10 respectively.

BLEU	translator ID	Set of Talk ID's
13.62	221131	1137, 1176, 1160, 1165
16.33	495543	1104, 1115, 1107
16.72	220760	1102, 1171

Table 9: `tst2011` scores for multiple talks by a single translator measured in BLEU.

The individual scores cover a surprising range; one question we want to explore is whether this reflects difficulty in the topic, expertise of the translator, or some combination of these. In Table 10, we see that a single translator can have a wide range of BLEU scores over different talks.

Next, we looked at the distribution of these translators in the training data. For the translators who did at least two talks in the test sets (`tst2010` through `tst2014`), we found that some had translated only a few of the training documents, while others had translated twenty or more documents as shown in Table 11.

While there is not enough training data by translator to train an entire system, there is enough data to try to create an MT system that is tuned to a particular translator. We compared a system adapted from System 0 from Table 4 with a variant system in which we held out training files from a particular translator to use as a dev set.

This creates four possibilities, as shown in Table 12. We can train on the original training files, or hold out the training documents by the specified translator; we can tune on `dev2010`, or on the held out training documents. We tested this approach using translator 495543 and translator 354776. Translator 495543 had three talks in `tst2011` that scored well in the original system; translator 364776 had 1 talk in `tst2011` that

BLEU	talkid	URL	translator ID
20.65	1104	eythor_bender_demos_human_exoskeletons	495543
9.95	1096	mark_bezos_a_life_lesson_from_a_volunteer_firefighter	193561
16.66	1102	isabel_behncke_evolution_s_gift_of_play_from...	220760
12.36	1166	alice_dreger_is_anatomy_destiny	831361
11.21	1161	jessi_arrington_wearing_nothing_new	925579
11.64	1137	carlo_ratti_architecture_that_senses_and_responds	221131
15.57	1171	camille_seaman_haunting_photos_of_ice	220760
17.01	1115	mick_ebeling_the_invention_that_unlocked_a_locked...	495543
17.24	1176	jok_church_a_circle_of_caring	221131
13.53	1107	ralph_langner_cracking_stuxnet_a_21st_century...	495543
10.02	1114	morgan_spurlock_the_greatest_ted_talk_ever_sold	354776
15.44	1144	amit_sood_building_a_museum_of_museums_on_the...	250727
16.80	1160	aaron_o_connell_making_sense_of_a_visible_quantum...	221131
12.15	1165	paul_romer_the_world_s_first_charter_city	221131

Table 10: `tst2011` per-talk scores measured in cased BLEU.

	Test		Train	
translator ID	docs	lines	docs	lines
221131	4	344	12	1346
1077318	2	330	4	429
495543	3	235	29	3248
250727	3	192	34	3995
1636197	2	167	26	2232
1648682	2	167	7	1022
1213653	2	147	21	2046
1053094	2	122	14	1806
220760	2	84	41	4373

Table 11: Distribution of Translator effort across test and train sets.

docs 1104, 1115, 1107;	**translator=495543**
dev2010	dev=495543
train (all) 15.89	15.29
train – 495543 13.29	14.08
docs 1114;	**translator=354776**
dev2010	dev=354776
train (all) 9.89	8.64
train – 354776 8.77	8.99

Table 12: Effect of translator-specific tuning on scores for specified `tst2011` documents reported in cased BLEU.

	dev2010	dev=495543	dev=354776
train (all)	16.70	13.39	11.25
train – 495543	13.94	14.90	–
train – 354776	14.84	–	15.10

Table 13: Effect of translator-specific tuning on scores on full `tst2011` reported in cased BLEU.

scored poorly in the original system. The held-out data for translator 495433 had 3248 lines; the held-out data for translator 354776 had 5335 lines.

When training on the restricted training data, we see an improvement for both translators in tuning on the held out data instead of `dev2010`. However, this tuning improvement is not enough to offset an overall drop in score from the reduction in training data.

Looking at scores for the complete `tst2011` test set, shown in Table 13, we see an expected drop in BLEU when restricting the training data (Column 1) and we continue to see an improvement in score when tuning on one of the `tst2011` translators instead of tuning with `dev2010` (Rows 2 and 3). Similar improvements with translator-specific tuning were seen for `dev2010`, `tst2012`, and `tst2013`, even though those test sets do not contain talks by these particular translators.

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 26 Oct 2015. Originator Reference Number: RH-15-114705. Case Number: 88ABW-2015-5214.

6. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2015 Evaluation Campaign," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT'15)*, ser. Proceedings of IWSLT, 2015.

[2] M. Cettolo, C. Girardi, and M. Federico, "WIT3:

Web Inventory of Transcribed and Translated Talks," ser. Proceedings of EAMT, 2012, pp. 261–268.

[3] M. Kazi, E. Salesky, B. Thompson, J. Ray, M. Coury, T. Shen, Wade Anderson, G. Erdmann, J. Gwinnup, K. Young, B. Ore, and M. Hutt, "The MIT-LL/AFRL IWSLT-2014 MT system," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT'14)*, Lake Tahoe, California, December 2014.

[4] P.-C. Chang, M. Galley, and D. C. Manning, *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2008, ch. Optimizing Chinese Word Segmentation for Machine Translation Performance, pp. 224–232.

[5] W. A. Gale, "Good-turing smoothing without tears," *Journal of Quantitative Linguistics*, vol. 2, 1995.

[6] J. Gwinnup, T. Anderson, G. Erdmann, K. Young, C. May, M. Kazi, E. Salesky, and B. Thompson, "The AFRL-MITLL WMT15 system: There's more than one way to decode it!" in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 112–119. [Online]. Available: http://aclweb.org/anthology/W15-3011

[7] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

[8] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[9] K. Heafield, P. Koehn, and A. Lavie, "Grouping language model boundary words to speed k-best extraction from hypergraphs," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA, June 2013, pp. 958–968.

[10] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08, 2008, pp. 848–856.

[11] M. Junczys-Dowmunt, "Phrasal rank-encoding: Exploiting phrase redundancy and translational relations for phrase table compression," *The Prague Bulletin of Mathematical Linguistics*, vol. 98, pp. 63–74, 2012.

[12] G. Erdmann and J. Gwinnup, "Drem: The AFRL submission to the WMT15 tuning task," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 422–427. [Online]. Available: http://aclweb.org/anthology/W15-3054

[13] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for training large scale neural network language models," ser. Automatic Speech Recognition and Understanding Workshop, 2011.

[14] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[15] I. Sutskever, O. Vinyals, and Q. V. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[16] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 355–362.

[17] J. Devlin, C. Quirk, and A. Menezes, "Pre-computable multi-layer neural network language models," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 256–260. [Online]. Available: http://aclweb.org/anthology/D15-1029

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[19] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[20] A.-V. I. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz, "Incremental hypothesis alignment for building confusion networks with application to machine translation system combination," in *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2008, pp. 183–186.

[21] M. Kazi, M. Coury, E. Salesky, J. Ray, W. Shen, T. Gleason, T. Anderson, G. Erdmann, L. Schwartz, B. Ore, R. Slyh, J. Gwinnup, K. Young, and M. Hutt, "The MIT-LL/AFRL IWSLT-2013 MT system," in *The 10th International Workshop on Spoken Language Translation (IWSLT '13)*, Heidelberg, Germany, December 2013, pp. 136–143.

[22] A. Larcher, J.-F. Bonastre, B. Fauve, K. A. Lee, C. Levy, H. Li, J. Mason, and J.-Y. Parfait, "ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition," in *Proceedings of Interspeech*, Lyon, France, August 2013.

[23] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 30–42, January 2012.

[24] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker Adaptation of Neural Network Acoustic Models using I-Vectors," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, December 2013.

[25] S. Zhang, H. Jiang, S. Wei, and L.-R. Dai, "Rectified Linear Neural Networks with Tied-Scalar Regularization for LVCSR," in *Proceedings of Interspeech*, Dresden, Germany, September 2015.

[26] P. Shen, X. Lu, X. Hu, N. Kanda, M. Saiko, and Chiori, "The NICT ASR system for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT'14)*, Lake Tahoe, California, December 2014.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011.

[28] F. Grezl and P. Fousek, "Optimizing Bottle-Neck Features for LVCSR," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[29] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[30] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *International Workshop on Spoken Language Translation*, Pittsburgh, PA, USA, Oct. 2005, pp. 148–154.