Abstract

This report summarizes the MITLL-AFRL MT and ASR systems and the experiments run during the 2016 IWSLT evaluation campaign. Building on lessons learned from previous years’ results, we refine our ASR systems and examine the explosion of neural machine translation systems and techniques developed in the past year. We experiment with a variety of phrase-based, hierarchical and neural-network approaches in machine translation and utilize system combination to create a composite system with the best characteristics of all MT approaches.

Preliminary results are denoted by * in this draft.

1. Introduction

During the evaluation campaign for the 2016 International Workshop on Spoken Language Translation (IWSLT16) [1] our experimental efforts in machine translation (MT) focused on the extension of our efforts from WMT16[2] and IWSLT15[3] and the exploration of many new neural machine translation (NMT) techniques including the refinement and improvement of our in-house NMT techniques, advanced selection techniques for parallel training data and the combination of this myriad of systems and techniques via system combination.

Our Automatic Speech Recognition (ASR) systems largely remain the same as last year, with the exception of training with this year’s additional data.

2. Machine Translation

For our efforts in the machine translation task this year we acknowledge the recent explosion of neural machine translation (NMT) techniques and leverage our previous experience with phrase-based and hierarchical machine translation systems to create a best-of-breed machine translation system via the technique of system combination.
Arabic words on different lines. We used line-final punctuation as a guide to assemble English lines into full sentences, while simultaneously concatenating their Arabic counterparts. Some Arabic files contain lines with just a period, corresponding to a blank line in the English file; we removed over 800 of these placeholders during concatenation. Our concatenation process failed when the sentences lacked punctuation or when the sentence-final punctuation fell in the middle of the line. This type of data led to very long concatenated sections. We therefore excluded files which exhibited excessive concatenation, measured as either a series of 5 or more concatenations with more than 500 total characters, or as an overall average of 30 or more words per line. These restrictions excluded 330 of our concatenated files. Three files were blank, and one other file was excluded on the basis of extremely bad spelling (including lowercase letter l for the personal pronoun, 1). We retained 889 out of 1223 files, for a total of 72,475 concatenated lines.

The assembly of QED transcripts from short video segments may also lead to a chunking error, in which words are run together in the middle of each line in the file. We observed this type of error in a small number of English files. We used the Aspell\(^1\) spell checker to identify run-together words such as andthe where the whole word is not in the Aspell dictionary, but the component words can be found. We had to manually review the list of suggested corrections to prevent the splitting of unknown names and technical terms; these were added to a supplemental Aspell dictionary. We also had to correct our automatic splits in some cases where there were multiple ways to split a run-together word (e.g., breadcrumbscan → breadcrumb scan OR breadcrumbs can). We implemented our corrections via table entries for the Varcon\(^2\) variant conversion program.

Spelling was particularly bad in some English talks in the corpus. For example, completly, enviornment, actualy, regardless, satilites, correspingding, pricipal, and so on. Many misspelled words showed up during our manual review of chunking errors. We identified files with excessive spelling problems and created additional spelling correction entries for the Varcon tables.

### 2.3. Training Data Subselection

Using definitions below, we select as a parallel training set a subset \(S\) from a large, general set \(C\) to maximize its similarity to a target set \(T\), using a coverage metric \(g(S,T)\). Defining \(c_i(X)\) as the count of feature \(i\)’s occurrence in corpus \(X\),

\[
g(S, T) = \frac{\sum_{i \in T} f(\min(c_i(S), c_i(T)))}{\sum_{i \in T} f(c_i(S)) + p_i(S, T)}
\]

where the over-saturation penalty \(p_i(S,T)\) is

\[
\max(0, c_i(S) - c_i(T)) [f(c_i(T) + 1) - f(c_i(T))].
\]

The coverage maximization problem, \(\max_{S \subseteq C} g(S,T)\), is solved via greedy optimization, iteratively adding the segment to \(S\) that provides the largest increase in \(g\). The set \(S\) is reviewed after each addition, removing any older segment in \(S\) that decreases \(g\).

We use \(f(x) = \log(1 + x)\) as a submodular function to weight counts. For Arabic the feature set \(Z\) is composed of all unigrams and bigrams, based on testing over \(n\)-gram lengths. For English trigrams are added, again based on empirical testing. In our usage the set \(C\) is the Parallel UN corpus, and the target set \(T\) is composed of the TED dev and test sets from 2010–2013.

### 2.4. Neural Probabilistic Language Model Experiments

We trained several Neural Probabilistic Language Models (NPLM), partly with the goal of seeing whether the gain from hybrid neural MT systems was clearly better than augmenting a phrase-based system with feedforward networks. We also intended to try a character-level version of the Devlin [12] Neural Network Joint Model (NNJM). The character-input version replaces the input word vector layer with the convolutional approach described in [13]. To this end, we trained our own Tensorflow \([14]\) implementation, and output the network in the NPLM format as required by Moses. We trained the model using the standard source context of 11, target of 3, and one to two hidden layers of size 512. The model was trained on in-domain TED data and validated on tst2014. The NNJM results are indicated in Table 1.

<table>
<thead>
<tr>
<th>NNJM Description</th>
<th>Cased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline, mosestoken</td>
<td>27.42</td>
</tr>
<tr>
<td>NNJM 2 HL, Rescoring</td>
<td>27.83</td>
</tr>
<tr>
<td>NNJM 2 HL x4 (s2,t2s,12,r2)</td>
<td>28.12</td>
</tr>
<tr>
<td>NNJM 1 HL, Decoding</td>
<td>27.61</td>
</tr>
<tr>
<td>Character 2 HL, Rescoring</td>
<td>27.75*</td>
</tr>
<tr>
<td>Character 1 HL, Decoding</td>
<td>27.75*</td>
</tr>
</tbody>
</table>

Table 1: Effects of NNJM integration into a baseline system without special Arabic processing, showing the benefit of character-level features on unstemmed data. Results are shown in BLEU decoding tst2014.

### 2.5. Moses MT Systems

Our baseline phrase-based system used the standard Moses [15] toolkit and only the provided in-domain training data. All Moses systems were tuned with Drem[16]. This baseline system was utilized as System 3 in system combination, shown in Table 5. Utilizing the parallel data selected in §2.3, we trained an additional system for combination listed as System 5.

While training additional Moses systems, for the purposes of obtaining an aligned development set (for Devlin models), we ran GIZA[17] on the in-domain TED data as

\[^{1}\text{http://aspell.net}\]

\[^{2}\text{http://wordlist.aspell.net/varcon}\]
well as tst2012, but only using the former to build phrase-tables and models. The resulting improvement in GIZA alignment quality does make a small difference in translation quality. Our in-domain system additionally employed truecasing (trained on TED), hierarchical MSLR reordering [18], order 5 operational sequence model [19], an order-7 word class in-domain language model, and a 6-gram in-domain language model.

In addition to in-domain TED data, we used our language model from WMT16 consisting of all of the newscrawl data from 2007-2014, plus the news discussions and Europarl corpora. For extra parallel data, we experimented with domain adaptation from Multi-UN, Parallel UN, QED, and OpenSubtitles corpora. For the selection process, we used bilingual cross-entropy data selection [20], specifically the latest method from Axelrod et al [21], where we replace words outside of the top 10K most frequent words by a tag that includes the part-of-speech and the relative frequency of the word in the in-domain versus out-of-domain datasets. For the English data, we used the Stanford Part-of-Speech tagger [22], and for the Arabic, we induced word classes with ClusterCat[23]. The frequency bins used were powers of 10, as in [21]. We achieved a significant gain using Multi-UN data, as can be seen in Table 2.

### Table 2: Effects of additional parallel training data for phrase-based MT as scored against tst2014.

<table>
<thead>
<tr>
<th>Dataset + Num selected</th>
<th>Separate PT</th>
<th>Combined PT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>-</td>
<td>27.42</td>
</tr>
<tr>
<td>multi UN 500K</td>
<td>27.58</td>
<td>27.81</td>
</tr>
<tr>
<td>multi UN 1M</td>
<td>27.76</td>
<td>27.71</td>
</tr>
<tr>
<td>UN v1.0 1M</td>
<td>27.61</td>
<td>27.72</td>
</tr>
<tr>
<td>QED all</td>
<td>27.23</td>
<td>27.63</td>
</tr>
<tr>
<td>OpenSubtitles 1M</td>
<td>27.23</td>
<td>26.88</td>
</tr>
<tr>
<td>OpenSubtitles 2M</td>
<td>27.27</td>
<td>23.68</td>
</tr>
</tbody>
</table>

This same model was used to rerank the scores from our best phrase-based system, boosting the scores from 27.90 to 28.90 on tst2014.

Using this same technique for the QED task, we fine-tuned the same model on the QED training set described in Section 2.2.3, validating on a dev set (comprised of talks 0cvcHoFwiJxVO, eODkKYZ2cmjf, fbpZ98nxEgnj, SFFR5jvxTZHl, T4hMt9Ft5CKP, WVL2qxNoFDCc, Z6SoWj12G6Em, ZPAQGyVEAUsV), boosting those scores by a factor of nearly double (See Table 7).

Since our phrase-based system did not use byte-pair encoding, the rescoring of the n-best list had a preprocessing step. This also made it difficult to decode with the model (see §2.8), so for the hybrid system, we also trained a Nematus system on truecase but not byte-pair encoded data, using an input vocabulary size of 160K and an output vocabulary of 80K. Decoding tst2014, this system achieved approximately 27 BLEU in the initial pass, and fine-tuned to 27.5* BLEU.

### 2.6. Nematus Systems

We were able to successfully train multiple Nematus [24] systems to achieve results on-par or slightly better than our phrase-based systems. Using the WMT16 scripts provided by the author, we trained a system using all of the Multi-UN data. The system used byte-pair encoding trained on the union of Arabic and English text, with 160,000 split operations. The resulting vocabularies were approximately 120K source tokens and 80K target tokens. We validated this model during training on IWSLT tst2012 until the scores stabilized at approximately 29 uncased BLEU. Then we fine-tuned the model using the in-domain TED dataset, for a small number of epochs. This achieved 34.21* BLEU on tst2012 and 28.1 cased-BLEU on tst2014 - this system is utilized as System 8 in system combination as shown in Table 5.

As a contrast, we also trained a Nematus system that used Farasa [10] to create subword units on the Arabic side, and byte-pair encoding on the English side. This system utilized a vocabulary size of 120,000 combined source/target. This system was motivated by the fact that Arabic words have prefixes and suffixes added to a root word to denote morphological information. By breaking apart the root word and the morphological suffixes and prefixes, we can reduce the size of the vocabulary used by a NMT system to the root words and a common set of prefixes and suffixes. While not performing as well as our straight byte-pair encoding Nematus system, it does add diversity to the systems used in system combination as described in §2.9. The result of decoding with the single-best model is listed as System 1 and the result of ensemble decoding with the 8-best models is listed as System 2.

### 2.7. Lamtram Systems

Following the success of the Nematus models, we trained additional neural machine translation systems using Lamtram [25]. We used 2x200 dimensional hidden layers. Our best system utilized the MultiUN corpus with byte-pair encoding and post-trained using the TED training data. Lamtram can incorporate probabilities from an external lexicon to boost translation probabilities. We used fast_align [26] to generate an IBM Model 1 lexicon. We did not have success running with minimum-risk. Results stabilized after 4 epochs with the UN data. BLEU scores improved by increasing beam size up to width of 10 which was the maximum possible under our GPU device contraints. Results are shown in Table 3. The best system was used in as System 4 in system combination (Table 5).
## 2.8. Hybrid MT Systems

An exciting development was the integration of neural MT models directly into the decoder – which is made possible by running on GPUs, using the AMUNMT\[27\] Moses variant. We used this tool to decode with our Nematus trained systems, yielding the improvements indicated in Table 4. We received more benefit from rescoring with the byte-pair encoded model than decoding with the non-byte-pair encoded model, and saw no gain from using both of them simultaneously.

<table>
<thead>
<tr>
<th>System Combination</th>
<th>Cased BLEU</th>
<th>Uncased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase-based BPE</td>
<td>26.92</td>
<td></td>
</tr>
<tr>
<td>+ NMT Decoding</td>
<td>27.89</td>
<td></td>
</tr>
<tr>
<td>Phrase-based no BPE</td>
<td>27.42</td>
<td></td>
</tr>
<tr>
<td>+ NMT Decoding</td>
<td>28.04</td>
<td></td>
</tr>
<tr>
<td>+/- NMT Rescoring BPE</td>
<td>29.10</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Hybrid PB/NMT results decoding tst\(2014\) reported in cased BLEU

## 2.9. System Combination

With the wide variety of systems and techniques tested this year, system combination becomes important. We examined methods to combine the disparate translation outputs. Inspired by the success of the combination of multiple systems in the QT21/HimL submission\[28\] to WMT16\[8\], we utilized RWTH’s Jane system combination technique \[29\] to combine outputs from each system to produce a unified, better translation result. Individual system inputs and combination results for decoding tst\(2014\) are listed in Table 5.

To determine the relative similarity of different system outputs for the purpose of system combination, we used automatic evaluation metrics. Outputs were scored against each other, using one of the outputs as the “reference”. Figure 1 shows a comparison between systems, based on output for tst\(2014\). Many metrics, such as BLEU, are asymmetric. The row of the table identifies the corpus treated as the hypothesis, and the column is the corpus treated as the reference. We see that the NMT systems are similar to each other, as are non-NMT systems. From experience we expect the greatest gain from system combination will be from combining somewhat dissimilar systems. Thus, we used at least one NMT and at least one non-NMT system in the final combination set.

In fact, each of the 10 systems used in our system combination are very different (with the exception of systems 5 and 6, which are preprocessed and trained the same aside from decoding strategy), even more so than the various component systems for QT21/HimL\[28\] submission to WMT16 (with the exceptions of closely-related systems 1-2 and 9-10.)

In Figure 2 we see the results of comparing the system-combination output against the next-best scoring contributing system utilizing the MTComparEval tool \[30\].

### 2.10. MT Results

We ultimately submitted two systems for evaluation: our best individual system-combination effort and our best phrase-based system with neural MT rescoring. These systems are listed in Table 6.

<table>
<thead>
<tr>
<th>System Combination</th>
<th>BLEU</th>
<th>Cased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Combination</td>
<td>30.49</td>
<td>29.41</td>
</tr>
<tr>
<td>Moses + NemResc. + UNPT</td>
<td>29.41</td>
<td>30.13</td>
</tr>
</tbody>
</table>

Table 6: Submission systems scores reported in BLEU decoding tst\(2014\).

Table 7 shows the individual contributions of the methods used in our phrase-based + neural rescoring submission system, also noting the contributions of the cleaned QED corpus shown in §2.2.3.

### 3. ASR

ASR systems were trained and evaluated using the same procedure as in IWSLT 2015 \[3\], except that this year we used...
Table 8 shows the word error rate (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. The final hypothesis for each utterance was selected by applying N-best Recognizer Output Voting Error Reduction (ROVER) to the output from the HTK adapted system and the Kaldi bottleneck system. The combined system yielded an 8.6% WER on tst2013 and an 8.9%* WER on tst2016.

3.1. QED Corpus for ASR Language Model

The English QED files were processed to correct chunking errors (see Section 2.2). We corrected run-together words in 57 files. In addition, 4 files were excluded on the basis of significant problems with spelling or foreign language sections. One file was also excluded due to problems with duplicated and partially duplicated lines.
Table 7: Additive scores for Moses + Nematus rescore system submission on t.st2014 (unless otherwise noted) as measured in BLEU.

<table>
<thead>
<tr>
<th>System</th>
<th>Cased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>25.02</td>
</tr>
<tr>
<td>+ DREM</td>
<td>25.55</td>
</tr>
<tr>
<td>+ Newscrawl LM</td>
<td>27.42</td>
</tr>
<tr>
<td>+ UN data</td>
<td>27.81</td>
</tr>
<tr>
<td>+ Rescore Nematus</td>
<td>29.41</td>
</tr>
<tr>
<td>Baseline UN Nematus</td>
<td></td>
</tr>
<tr>
<td>+ Finetune TED</td>
<td>28.10</td>
</tr>
<tr>
<td>+ Ensemble</td>
<td>28.53*</td>
</tr>
</tbody>
</table>

Table 8: English t.st2013 WER.

<table>
<thead>
<tr>
<th>ASR System</th>
<th>Decode</th>
<th>4-gram</th>
<th>4-gram+RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTK first-pass</td>
<td>12.8</td>
<td>12.2</td>
<td>10.8</td>
</tr>
<tr>
<td>HTK adapted</td>
<td>10.8</td>
<td>10.4</td>
<td>9.5</td>
</tr>
<tr>
<td>Kaldi bottleneck</td>
<td>11.7</td>
<td>11.3</td>
<td>10.4</td>
</tr>
</tbody>
</table>

4. Conclusion

In closing we see that neural machine translation systems are the new, exciting area of research in the problem-space of machine translation despite growing pains. We see that the “old wisdom” of statistical machine translation systems is still useful and that a thoughtful combination of the two can produce translations greater than the sum of their parts.

5. References


