Abstract
This paper describes the AFRL-MITLL statistical machine translation systems and the improvements that were developed during the WMT16 evaluation campaign. As part of these efforts we have adapted a variety new techniques to our previous years’ systems including Neural Machine Translation, additional out-of-vocabulary transliteration techniques, and morphology generation.

1 Introduction
As part of the 2016 Conference on Machine Translation (WMT16) news-translation shared task, the MITLL and AFRL human language technology teams participated in the Russian–English and English–Russian news translation tasks. Our machine translation (MT) systems represent improvements to both our systems from IWSLT2015 (Kazi et al., 2015) and WMT15 (Gwinnup et al., 2015), the introduction of Neural Machine Translation rescoring, neural-net based recasing, unsupervised transliteration of out-of-vocabulary (OOV) words (Durrani et al., 2014), and an unique selection process for language modelling data. For the English–Russian translation task we experimented with morphology generation techniques to improve translation quality.

2 System Description
We submitted systems for the Russian–English and English–Russian news-domain machine translation shared tasks. For all submissions, we used the phrase-based variant of the moses decoder (Koehn et al., 2007). In some cases we used a performance-enhanced version of Moses. As in previous years, our submitted systems used only the constrained data supplied when training.

2.1 Data Usage
In training our systems we utilized the following corpora to train translation and language models: Yandex1, Common Crawl (Smith et al., 2013), LDC Gigaword English v5 (Parker et al., 2011) and News Commentary. For additional language modelling data we processed the new Common Crawl monolingual corpora using the techniques described in §2.4. The Wikipedia Headlines corpus2 was reserved to train named entity recognizers.

2.2 Data Preprocessing
We processed the training data similarly to our WMT15 system (Gwinnup et al., 2015). We examined irregular behaviors in Moses’s punctuation normalization script3. We ran a script that examines the source and target side of the parallel training data and removes lines that are identical in both the source and target in order to prevent the effects of wrong-language phrases “polluting” the phrase and rule tables.

2.3 Phrase Table Generation
We used the standard Moses method of extracting and creating phrase tables. Phrase tables were binarized using either the Compact Phrase Table (Junczys-Dowmunt, 2012) or ProbingPT (?) methods.

2.4 Language Model Data Selection
Using definitions below, we select as a language modelling set a subset $S$ from the Common Crawl

---

1https://translate.yandex.ru/corpus?lang=en
2http://statmt.org/wmt15/wiki-titles.tgz
3normalize-punctuation.perl
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 $X$, $g(S, T) = \sum_{i \in \mathcal{I}} f(\min(c_i(S), c_i(T))) / \sum_{i \in \mathcal{I}} f(c_i(T)) + p_i(S, T)$ where the oversaturation penalty $p_i(S, T)$ is $\max(0, c_i(S) - c_i(T)) \left[ f(c_i(T) + 1) - f(c_i(T)) \right]$. We use $f(x) = \log(1 + x)$ as the submodular function to weight counts, and the feature set $\mathcal{I}$ is the set of all unigrams and bigrams. The target set $T$ is made of the news test sets from 2013–2015.

The optimization 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 other segment in $S$ that decreases $g$.

The Common Crawl corpus $C$ is broken into easily-processed chunks of ten thousand segments, selecting five hundred segments from each chunk. This selection was repeated until we saw diminishing returns from adding further chunks, resulting in a language modelling subset of six million lines. These six million lines represent 0.17% of the 3.6 billion lines of data in the English portion of the Common Crawl.

2.5 Tuning Improvements

Improvements were made to our tuner, Drem (Erdmann and Gwinnup, 2015), since our last submission. Enforcement of minimum and maximum distance of the tuning result from prior decodes (i.e., tabu and fear constraints) is now implicitly enforced via $L_1$ penalty functions, making the process more robust to densely-packed decodes. Rescoring weights are now not penalized in the n-best list interpolation scheme, since they do not directly affect n-best lists. This new feature provides faster convergence of our NMT-rescored systems. Another improvement to Drem is that the metric chrF3 (Popović, 2015) is now available as a tuning objective function.

2.6 Neural Network Recaser

We noticed a substantial gap between uncased and cased BLEU scores on our systems. Addressing the problem in post-processing, it became apparent that recasing can only do so much on monolingual data. We therefore built a classifier that uses both the source-side and the target-side of the translations. The inputs to the classifier are:

- $t_i$, the word to be recased, as well as $t_{i-1}$ and $t_{i-2}$
- $s_u(i)$, the source word aligned to $t_i$, plus $s_u(i) \pm 1$. Alignments were taken from Moses output, and missing alignments were computed using the NNJM affiliation heuristic (Devlin et al., 2014).
- The status of the source word as lowercase, capitalized, or OTHER.

The exact classifier used could be anything; we chose a neural network because it is simple to create and robust. Our architecture is as follows:

1. Vocabulary of all words, excluding 25% of singletons
2. Input: Word vectors for these words, plus nine binary inputs ($s_{i-1} = lc$, $s_{i-1} = uc$, $s_i = OTHER$, $s_i = lc$ \ldots), all concatenated together into a single vector
3. Two hidden layers, default size 100
4. One softmax output, 3 output classes

The resulting recaser consistently yields +0.2–0.25 case-sensitive BLEU over a standard language model recaser.

2.7 Inflection Generation

English-Russian systems have the added challenge of generating morphologically rich word-forms. In addition to an English-Russian baseline, we trained two methods to generate inflected forms. First, we created a system with a separate inflection prediction component (Toutanova et al. 2008, Fraser et al. 2012). We trained an MT system from English to lemmatized Russian, using the Mystem\textsuperscript{4} Russian morphological analyzer to lemmatize all available parallel data, and then trained a MT system from lemmatized Russian to Russian. Scoring against lemmatized references, the first step yielded 27.70 case-insensitive BLEU on newstest2016. However, while the lemru-ru system was successful with one-to-one lemmatized training data, it couldn’t recover from mistakes in the MT output of the first step and the system overall did not perform as well as our baseline (17.19 cased BLEU).

We also attempted to address inflection generation during training using verb annotation, following the approach of Kirchhoff et al. (2015) for Arabic verb inflection. We use dependency parsing to

\textsuperscript{4}https://api.yandex.ru/mystem
Wouldn’t you know it?

Annotated:  Would n’t you know-2p it ?

Dependency Parse:

<table>
<thead>
<tr>
<th>Index</th>
<th>Word</th>
<th>POS</th>
<th>Head</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Would</td>
<td>MD</td>
<td>4</td>
<td>aux</td>
</tr>
<tr>
<td>2</td>
<td>n’t</td>
<td>RB</td>
<td>4</td>
<td>neg</td>
</tr>
<tr>
<td>3</td>
<td>you</td>
<td>PRP</td>
<td>4</td>
<td>nsubj</td>
</tr>
<tr>
<td>4</td>
<td>know</td>
<td>VB</td>
<td>0</td>
<td>root</td>
</tr>
<tr>
<td>5</td>
<td>it</td>
<td>PRP</td>
<td>4</td>
<td>dobj</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td>.</td>
<td>4</td>
<td>punct</td>
</tr>
</tbody>
</table>

Figure 1: Annotation via Dependency Parse

identify the subject of the verb in the English sentence and then annotate the verb with the person and number of the subject. With a pronoun subject he or she, the verb is also annotated for gender. This provides the potential for the system to match annotated English verbs to the correctly inflected Russian verbs during training. Figure 1 shows an annotated sentence and the underlying dependency parse.

We use the Stanford parser (Klein and Manning, 2003) and conversion utility to generate the dependency parses, adjusting the tokenization of the input to match the Stanford treatment of contractions. We apply annotation to verbs with subjects listed as nsubj or xsubj in the dependency parse. Person, number, and gender are derived from the subject’s POS tag and from the specific lexical item for pronouns. Coordinate subjects are counted as plural.

An unannotated MT system has a good chance of associating the correct verb form with the subject if the subject and verb are adjacent and can be extracted as a phrase, while more distant pairs are less likely to be found in the phrase table, leaving the verb open to translation in the wrong inflected form. Since annotation can increase data sparsity, it is better to apply it only when necessary.

Kirchhoff et al. (2015) address the data sparsity issue by only applying their annotation-trained model when their baseline model translates the subject and verb via separate phrases. In some of our systems, we simulated the use of a back-off model by restricting our annotation to subjects and verbs that occur with a minimum separation distance.

Figure 2 shows the potential effect of specifying a minimum separation distance. In the first sentence, the subject and verb are adjacent; any separation requirement greater than zero prevents annotation of the verb. The other sentences show a greater separation, and annotation will be maintained if the separation requirement is less than 3.

In order to avoid the data sparsity problem, we ultimately created a factored version of the verb annotation system. The annotations were specified as factors on the verb, with a null factor on the unannotated words, e.g. would|NONE n’t|NONE you|NONE know|2p it|NONE ?|NONE

In system 2 of our English-Russian systems (shown in Table 8), we used this factored input with no separation limit.

2.7.1 Discussion

We examined the effect of verb annotation on inflection choice using an enhanced version of the Hjerson (Popović, 2011) error analysis program, in conjunction with the Mystem Russian morphological analyzer. Factored verb annotation as described above failed to reduce the number of inflectional errors (shown in Table 1.)

<table>
<thead>
<tr>
<th>Technique</th>
<th>Inf. Errors</th>
<th>Pct. Hyp. Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>5823</td>
<td>9.349%</td>
</tr>
<tr>
<td>Annotated</td>
<td>5994</td>
<td>9.351%</td>
</tr>
</tbody>
</table>

Table 1: Hjerson performance

The verb annotation technique aims to increase the information available for the generation of verb inflections. Errors in verb inflection amount to just a small proportion of overall errors in our baseline system, so the room for improvement in translation quality is small (shown in Table 2.)

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Instances</th>
<th>Pct. Hyp. Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Choice</td>
<td>30031</td>
<td>48.21%</td>
</tr>
<tr>
<td>Reordering</td>
<td>4479</td>
<td>7.19%</td>
</tr>
<tr>
<td>Inflection</td>
<td>5823</td>
<td>9.35%</td>
</tr>
</tbody>
</table>

Table 2: Hjerson classification of Error Types in Baseline System
Only about 18% of these 5823 baseline inflectional errors involve verbs; other errors involve nouns and pronouns (about 58%) or adjectives (about 24%). Meanwhile, the use of annotated data had unintended consequences for the other elements in the sentence. While our annotations were only applied to verbs in the training data, changes in inflection were observed for nouns and pronouns as well.

We used Mystem to provide a morphological analysis of the inflectional errors. We found that similar errors were made in both the baseline system and the annotated system. Looking at the error types by part of speech, we saw that verb errors for both systems primarily involved either number or gender, as opposed to tense or person. Pronoun errors for both systems showed a tendency for oblique cases in place of nominative.

For example, both systems displayed errors in which 

\[ \text{бьют (third person plural) "they will" was generated instead of the reference form, будет (third person singular) "he will"}. \]

The baseline system had 8 instances of this error, while the annotated system had 10 instances. The most frequent error was the substitution of the dative/locative first person singular pronoun мне “to me” for the nominative pronoun я “I”. The baseline system had 16 instances of this error, compared to 20 instances for the annotated system.

The verb-annotated system performed worse than our baseline when evaluated with the BLEU metric. We hope to gain more insight from the human ranking of the two systems.

### 2.8 Transliteration

We employed two methods to address transliteration of remaining out-of-vocabulary (OOV) words: an unsupervised statistical transliteration approach and a novel character-based neural-network transliteration approach.

#### 2.8.1 Neural Network Transliteration

We created a list of 54k Named Entity (NE) pairs from the Common Crawl using transliteration mining (Gwinnup et al., 2015) and employed this list in building a neural network based transliterator. We trained an encoder-decoder LSTM network to produce characters in a target language given characters from a word in the source language. The network configuration was nearly the same as that in our NMT experiments, except the network was significantly smaller (hidden sizes of 100 and 200, with 1, 2, and 3 hidden layers) and had a beam of 5. A small (5k) subset of the data was held out for evaluation/tuning. Since Russian nouns use case inflections, multiple Russian word forms may map to a single English spelling. For this reason, we tried rescoring with a unigram language model trained on the monolingual data to help weight the correct English spelling of words that may have been seen in the language modelling data but were not in the phrase table. The LM’s unknown word probability was optimized on the validation set.

<table>
<thead>
<tr>
<th>System</th>
<th>Exact matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [0 edit distance]</td>
<td>23.1%</td>
</tr>
<tr>
<td>Single enc-dec</td>
<td>34.7%</td>
</tr>
<tr>
<td>Ensemble (6)</td>
<td>38.7%</td>
</tr>
<tr>
<td>Single enc-dec + LM rescore</td>
<td>42.5%</td>
</tr>
<tr>
<td>Ensemble (6) + LM rescore</td>
<td>45.8%</td>
</tr>
</tbody>
</table>

Table 3: Fraction of transliterations that match exactly, on validation set (subset of newstest2014)

We integrated this into our SMT pipeline through different backoff phrase tables. Unknown words from the dev and test sets were transliterated via beam search (beam and stack size of 5) using the final system in Table 3 to create phrase table entries. The results are in Table 4. Gains may seem modest, however, there are not that many OOV words in newstest2015 – only 817 total unknowns, 515 of which we attempted to transliterate (ASCII entries and Capitalized words). Despite this, gains are consistent.

<table>
<thead>
<tr>
<th>System</th>
<th>Cased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. drop unknowns</td>
<td>28.07</td>
</tr>
<tr>
<td>2. pass-through unknowns</td>
<td>27.85</td>
</tr>
<tr>
<td>3. ASCII entries in backoff PT</td>
<td>27.86</td>
</tr>
<tr>
<td>4. 3 + cased words LM match</td>
<td>28.20</td>
</tr>
<tr>
<td>5. 3 + all cased Cyrillic words</td>
<td>28.16</td>
</tr>
</tbody>
</table>

Table 4: Neural Transliteration via Backoff PTs

#### 2.8.2 Unsupervised Statistical Transliteration

As a contrast to our neural network transliteration approach, we also experimented with using the unsupervised statistical transliteration method (Durani et al., 2014) included in Moses. System 2 in Table 7 and both systems in Table 8 employ this strategy as a post-decode step.
2.9 Neural MT

We describe a Neural Machine Translation system we developed and our strategies to integrate this system into our machine translation framework.

2.9.1 System
We trained a neural encoder-decoder network (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015) using the attention model from (Vinyals et al., 2015) to perform neural machine translation (NMT). We trained the model using Adagrad (Duchi et al., 2011) and found it improved performance over the learning rate schedule proposed in (Luong et al., 2015). We also found it advantageous to use a larger source vocabulary (200k-500k words worked well). Each instance of the system comprised of two 1000-dim hidden layers, with beam and stack of 5. Our NMT results are shown in Table 5. They did not perform competitively with our SMT systems by themselves, however they were very useful in rescoring – others have noted before the benefits of neural model rescoring (Auli et al., 2013).

<table>
<thead>
<tr>
<th>System</th>
<th>Cased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Single model</td>
<td>21.00</td>
</tr>
<tr>
<td>2. Ensemble of 2</td>
<td>21.46</td>
</tr>
</tbody>
</table>

Table 5: Russian–English Neural MT Systems decoding newstest2015

2.9.2 Reranking
We compared two different ways of using the NMT system to augment our phrase-based system.

1. **Single set of weights** We augment the Moses n-best list with NMT scores for each sentence, and then tune the decode weights using Drem. We repeat this process 10 times, using the last weights to decode the test set and one-best calculation.

2. **Decode + rerank weights** We tune the decode weights using Drem, without the NMT scores. After 10 iterations, we merge the n-best lists together and compute NMT scores over the result. Then, we compute a second set of weights. To decode the test set, we pass the decode weights to Moses, augment the n-best list with NMT scores, and finally apply the one-best dot product using the second set of weights.

The first process produced scores of 27.22, and the second 27.92 (mteval, case+punc, newstest2015, average of 6).

3 Results

We submitted 2 Russian–English and 2 English–Russian systems for evaluation, each employing a different decoding strategy. Each system is described below. Automatically scored results reported in BLEU (Papineni et al., 2002) for our submission systems can be found in Table 7 for Russian–English and Table 8 for English–Russian.

Finally, as part of WMT16, the results of our submission systems were ranked by monolingual human judges against the machine translation output of other WMT16 participants. These judgments are reported in WMT (2016).

3.1 Russian–English

For both Russian–English system submissions, we reused the BigLM15 concept from our WMT15 submissions to build a monolithic language model from the following sources: Yandex5, CommonCrawl (Smith et al., 2013), LDC Gigaword English v5 (Parker et al., 2011) and News Commentary. Submission system 1 included the data selected from the large CommonCrawl corpus as outlined in §2.4, while submission system 2 used this data to build a separate, complementary language model.

For submission system 1, we used a standard phrase based approach with the following parameters/features: distortion-limit of 8, no reordering over punctuation, hierarchical mslr reordering model (Galley and Manning, 2008), order 7 operational sequence model (Durrani et al., 2011), and a factored language model over the NYT Gigaword corpus with 600 word classes. We incorporated our Tensorflow Neural MT system in via reranking, and applied transliteration as backoff phrase tables during decoding. Lowercased output was recased via neural network. A breakdown of scores for submission system one is indicated in Table 6.

For submission system 2, we used the same approach as system 1, removing the class-factored language model and utilizing both the BigLM used in our WMT15 systems and a secondary language model built from data selected from the monolingual CommonCrawl corpus as outlined in §2.4. While this system did use the same transliteration

5https://translate.yandex.ru/corpus?lang=en
<table>
<thead>
<tr>
<th>Features</th>
<th>Cased BLEU tst15</th>
</tr>
</thead>
<tbody>
<tr>
<td>pb + BigLM</td>
<td>27.09</td>
</tr>
<tr>
<td>+ nmt</td>
<td>27.92</td>
</tr>
<tr>
<td>+ cc LM data</td>
<td>28.07</td>
</tr>
<tr>
<td>+ translit</td>
<td>28.20</td>
</tr>
</tbody>
</table>

Table 6: Score breakdown for submission system 1, average of 6 runs on newstest2015.

backoff phrase tables to handle OOVs, due to different preprocessing methodologies, some OOVs still remained in the output. The Moses unsupervised statistical transliterator was applied as a post-process. Finally, the Moses statistical recaser was employed to recase the data before scoring.

3.2 English–Russian

Both English–Russian submission systems used a language model interpolated from individual models built from all available Russian data.

Submission system 1 is a standard baseline system employing hierarchical lexicalized reordering and an order 5 operation sequence model.

For submission system 2, we applied factored verb annotation on the training data to guide inflection choice, as outlined in §2.7. This system also employed hierarchical lexicalized reordering and an order 5 operation sequence model. While this system did not perform as well as system 1, we are interested to see the effect of this verb-annotation approach on the human-ranking portion of the evaluation.

Due to time and processing constraints we did not employ Neural Machine Translation rescoring in our English–Russian submission systems.

4 Conclusion

In conclusion, we present a series of improvements to our Russian–English and English–Russian machine translation systems which represent significant increases in machine translation quality.

Acknowledgements

We wish to thank the anonymous reviewers for their comments and insight.

References


<table>
<thead>
<tr>
<th>System</th>
<th>Cased BLEU</th>
<th>Uncased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. pb + NMT rescore + BigLM(inc. CC data) + Neural translit</td>
<td>27.6</td>
<td>28.8</td>
</tr>
<tr>
<td>2. pb (clean data) + NMT rescore + BigLM + CC subsel LM + Neural translit + Moses translit</td>
<td>27.0</td>
<td>28.4</td>
</tr>
</tbody>
</table>

Table 7: Russian–English MT Submission Systems decoding newstest2016

<table>
<thead>
<tr>
<th>System</th>
<th>Cased BLEU</th>
<th>Uncased BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. enru-pb</td>
<td>23.42</td>
<td>23.52</td>
</tr>
<tr>
<td>2. enru-pb-facvban0</td>
<td>20.90</td>
<td>21.00</td>
</tr>
</tbody>
</table>

Table 8: English–Russian MT Submission Systems decoding newstest2016


