
The University of Cambridge Russian-English System at WMT13

Juan Pino Aurelien Waite Tong Xiao
Adrià de Gispert Federico Flego William Byrne
Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK
{jmp84,aaw35,tx212,ad465,ff257,wjb31}@eng.cam.ac.uk

Abstract

This paper describes the University of Cambridge submission to the Eighth Workshop on Statistical Machine Translation. We report results for the Russian-English translation task. We use multiple segmentations for the Russian input language. We employ the Hadoop framework to extract rules. The decoder is HiFST, a hierarchical phrase-based decoder implemented using weighted finite-state transducers. Lattices are rescored with a higher order language model and minimum Bayes-risk objective.

1 Introduction

This paper describes the University of Cambridge system submission to the ACL 2013 Eighth Workshop on Statistical Machine Translation (WMT13). Our translation system is HiFST (Iglesias et al., 2009), a hierarchical phrase-based decoder that generates translation lattices directly. Decoding is guided by a CYK parser based on a synchronous context-free grammar induced from automatic word alignments (Chiang, 2007). The decoder is implemented with Weighted Finite State Transducers (WFSTs) using standard operations available in the OpenFst libraries (Al-lauzen et al., 2007). The use of WFSTs allows fast and efficient exploration of a vast translation search space, avoiding search errors in decoding. It also allows better integration with other steps in our translation pipeline such as 5-gram language model (LM) rescoring and lattice minimum Bayes-risk (LMBR) decoding (Blackwood, 2010).

We participate in the Russian-English translation shared task in the Russian-English direction. This is the first time we train and evaluate a system on this language pair. This paper describes the development of the system. The paper is organised as follows. Section 2 describes each step in the development of our system for submission, from pre-processing to post-processing and Section 3 presents and discusses results.

2 System Development

2.1 Pre-processing

We use all the Russian-English parallel data available in the constraint track. We filter out non Russian-English sentence pairs with the language-detection library. A sentence pair is filtered out if the language detector detects a different language with probability more than 0.999995 in either the source or the target. This discards 78543 sentence pairs. In addition, sentence pairs where the source sentence has no Russian character, defined by the Perl regular expression `\x0400-\x04ff`, are discarded. This further discards 19000 sentence pairs.

The Russian side of the parallel corpus is tokenised with the Stanford CoreNLP toolkit. The Stanford CoreNLP tokenised text is additionally segmented with Morfessor (Creutz and Lagus, 2007) and with the TreeTagger (Schmid, 1995). In the latter case, we replace each token by its stem followed by its part-of-speech. This offers various segmentations that can be taken advantage of in hypothesis combination: CoreNLP, CoreNLP+Morfessor and CoreNLP+TreeTagger. The English side of the parallel corpus is tokenised with a standard in-house tokeniser. Both sides of the parallel corpus are then lowercased, so mixed case is restored in post-processing.

Corpus statistics after filtering and for various segmentations are summarised in Table 1.

---

2 http://code.google.com/p/language-detection/
3 http://nlp.stanford.edu/software/corenlp.shtml
Table 1: Russian-English parallel corpus statistics for various segmentations.

<table>
<thead>
<tr>
<th>Lang</th>
<th>Segmentation</th>
<th># Tokens</th>
<th># Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RU</td>
<td>CoreNLP</td>
<td>47.4M</td>
<td>1.2M</td>
</tr>
<tr>
<td>RU</td>
<td>Morfessor</td>
<td>50.0M</td>
<td>0.4M</td>
</tr>
<tr>
<td>RU</td>
<td>TreeTagger</td>
<td>47.4M</td>
<td>1.5M</td>
</tr>
<tr>
<td>EN</td>
<td>Cambridge</td>
<td>50.4M</td>
<td>0.7M</td>
</tr>
</tbody>
</table>

2.2 Alignments

Parallel data is aligned using the MTTK toolkit (Deng and Byrne, 2008). We train a word-to-phrase HMM model with a maximum phrase length of 4 in both source-to-target and target-to-source directions. The final alignments are obtained by taking the union of alignments obtained in both directions.

2.3 Rule Extraction and Retrieval

A synchronous context-free grammar (Chiang, 2007) is extracted from the alignments. The constraints are set as in the original publication with the following exceptions:

- phrase-based rule maximum number of source words: 9
- maximum number of source element (terminal or nonterminal): 5
- maximum span for nonterminals: 10

Maximum likelihood estimates for the translation probabilities are computed using MapReduce. We use a custom Hadoop-based toolkit which implements method 3 of Dyer et al. (2008). Once computed, the model parameters are stored on disk in the HFile format (Pino et al., 2012) for fast querying. Rule extraction and feature computation takes about 2h30. The HFile format requires data to be stored in a key-value structure. For the key, we use shared source side of many rules. The value is a list of tuples containing the possible targets for the source key and the associated parameters of the full rule. The query set of keys for the test set is all possible source phrases (including nonterminals) found in the test set.

During HFile querying we add other features. These include IBM Model 1 (Brown et al., 1993) lexical probabilities. Loading these models in memory doesn’t fit well with the MapReduce model so lexical features are computed for each test set rather than for the entire parallel corpus.

The model parameters are stored in a client-server based architecture. The client process computes the probability of the rule by querying the server process for the Model 1 parameters. The server process stores the model parameters completely in memory so that parameters are served quickly. This architecture allows for many low-memory client processes across many machines.

2.4 Language Model

We used the KenLM toolkit (Heafield et al., 2013) to estimate separate 4-gram LMs with Kneser-Ney smoothing (Kneser and Ney, 1995), for each of the corpora listed in Tables 2 (self-explanatory abbreviations). The component models were then interpolated with the SRILM toolkit (Stolcke, 2002) to form a single LM for use in first-pass translation decoding. The interpolation weights were optimised for perplexity on the news-test2008, newstest2009 and newsyscomb2009 development sets. The weights reflect both the size of the component models and the genre of the corpus the component models are trained on, e.g. weights are larger for larger corpora in the news genre.

Table 2: Statistics for English monolingual corpora.

<table>
<thead>
<tr>
<th>Corpus</th>
<th># Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU + NC + UN + CzEng + Yx</td>
<td>652.5M</td>
</tr>
<tr>
<td>Giga + CC + Wiki</td>
<td>654.1M</td>
</tr>
<tr>
<td>News Crawl</td>
<td>1594.3M</td>
</tr>
<tr>
<td>afp</td>
<td>874.1M</td>
</tr>
<tr>
<td>apw</td>
<td>1429.3M</td>
</tr>
<tr>
<td>cna + wpb</td>
<td>66.4M</td>
</tr>
<tr>
<td>ltw</td>
<td>326.5M</td>
</tr>
<tr>
<td>nyt</td>
<td>1744.3M</td>
</tr>
<tr>
<td>xin</td>
<td>425.3M</td>
</tr>
<tr>
<td>Total</td>
<td>7766.9M</td>
</tr>
</tbody>
</table>

2.5 Decoding

For translation, we use the HiFST decoder (Iglesias et al., 2009). HiFST is a hierarchical decoder that builds target word lattices guided by a probabilistic synchronous context-free grammar. Assuming N to be the set of non-terminals and T the set of terminals or words, then we can define the grammar as a set \( R = \{ R \} \) of rules \( R : N \rightarrow \langle \gamma, \alpha \rangle / p \), where \( N \in N \), \( \gamma, \alpha \in \{ N \cup T \}^+ \) and \( p \) the rule score.
HiFST translates in three steps. The first step is a variant of the CYK algorithm (Chappelier and Rajman, 1998), in which we apply hypothesis recombination without pruning. Only the source language sentence is parsed using the corresponding source-side context-free grammar with rules $N \rightarrow \gamma$. Each cell in the CYK grid is specified by a non-terminal symbol and position: $(N, x, y)$, spanning $s_x^{x+y-1}$ on the source sentence $s_1...s_J$.

For the second step, we use a recursive algorithm to construct word lattices with all possible translations produced by the hierarchical rules. Construction proceeds by traversing the CYK grid along the back-pointers established in parsing. In each cell $(N, x, y)$ of the CYK grid, we build a target language word lattice $L(N, x, y)$ containing every translation of $s_x^{x+y-1}$ from every derivation headed by $N$. For efficiency, this lattice can use pointers to lattices on other cells of the grid.

In the third step, we apply the word-based LM via standard WFST composition with failure transitions, and perform likelihood-based pruning (Allauzen et al., 2007) based on the combined translation and LM scores.

No alignment information is used when computing lexical scores as done in Equation (4) in (Koehn et al., 2005). Instead, the source-to-target lexical score is computed in Equation 1:

$$s(\mathbf{ru}, \mathbf{en}) = \frac{1}{(E + 1)^R} \prod_{r=1}^{R} \sum_{e=0}^{E} p_{M1}(\mathbf{en}_e | \mathbf{ru}_r)$$

(1)

where $\mathbf{ru}$ are the terminals in the Russian side of a rule, $\mathbf{en}$ are the terminals in the English side of a rule, including the null word, $R$ is the number of Russian terminals, $E$ is the number of English terminals and $p_{M1}$ is the IBM Model 1 probability.

In addition to these standard features, we also use provenance features (Chiang et al., 2011). The parallel data is divided into four subcorpora: the Common Crawl (CC) corpus, the News Commentary (NC) corpus, the Yandex (Yx) corpus and the Wiki Headlines (Wiki) corpus. For each of these subcorpora, source-to-target and target-to-source translation and lexical scores are computed. This requires computing IBM Model 1 for each subcorpus. In total, there are 28 features, 12 standard features and 16 provenance features.

When retrieving relevant rules for a particular test set, various thresholds are applied, such as number of targets per source or translation probability cutoffs. Thresholds involving source-to-target translation scores are applied separately for each provenance and the union of all surviving rules for each provenance is kept. This strategy gives slight gains over using thresholds only for the general translation table.

We use an implementation of lattice minimum error rate training (Macherey et al., 2008) to optimise under the BLEU score (Papineni et al., 2001) the feature weights with respect to the odd sentences of the newstest2012 development set (newstest2012.tune). The weights obtained match our expectation, for example, the source-to-target translation feature weight is higher for the NC corpus than for other corpora since we are translating news.

2.7 Lattice Rescoring

The HiFST decoder is set to directly generate large translation lattices encoding many alternative translation hypotheses. These first-pass lattices are rescored with second-pass higher-order LMs prior to LMBR.
2.7.1 5-gram LM Lattice Rescoring
We build a sentence-specific, zero-cutoff stupid-
backoff (Brants et al., 2007) 5-gram LMs esti-
mated over the data described in section 2.4. Lat-
tices obtained by first-pass decoding are rescored
with this 5-gram LM (Blackwood, 2010).

2.7.2 LMBR Decoding
Minimum Bayes-risk decoding (Kumar and
Byrne, 2004) over the full evidence space of the 5-
gram rescored lattices is applied to select the trans-
lation hypothesis that maximises the conditional
expected gain under the linearised sentence-level
BLEU score (Tromble et al., 2008; Blackwood,
2010). The unigram precision $p$ and average re-
call ratio $r$ are set as described in Tromble et al.
(2008) using the newstest2012.tune development
set.

2.8 Hypothesis Combination
LMBR decoding (Tromble et al., 2008) can also be
used as an effective framework for multiple lattice
combination (Blackwood, 2010). We used LMBR
to combine translation lattices produced by sys-
tems trained on alternative segmentations.

2.9 Post-processing
Training data is lowercased, so we apply true-
casing as post-processing. We used the disam-
big tool provided by the SRILM toolkit (Stolcke,
2002). The word mapping model which contains
the probability of mapping a lower-cased word
to its mixed-cased form is trained on all avail-
able data. A Kneser-Ney smoothed 4-gram lan-
guage model is also trained on the following cor-
pora: NC, News Crawl, Wiki, afp, apw, cna, ltw,
nyt, wpb, xin, giga. In addition, several rules are
manually designed to improve upon the output of
the disambig tool. First, casing information from
pass-through translation rules (for OOV source
words) is used to modify the casing of the output.
For example, this allows us to get the correct cas-
ing for the word Bundesrechnungshof. Other rules
are post-editing rules which force some words
to their upper-case forms, such as euro → Euro.
Post-editing rules are developed based on high-
frequency errors on the newstest2012.tune develop-
ment set. These rules give an improvement of
0.2 mixed-cased NIST BLEU on the development
set.

Finally, the output is detokenised before sub-
mission and Cyrillic characters are transliterated.

3 Results and Discussion
Results are reported in Table 3. We use the inter-
nationalisation switch for the NIST BLEU scor-
ing script in order to properly lowercase the hy-
pothesis and the reference. This introduces a
slight discrepancy with official results going into
the English language. The newstest2012.test de-
velopment set consists of even sentences from
newstest2012. We observe that the CoreNLP
system (A) outperforms the other two systems.
The CoreNLP+Morfessor system (B) has a much
smaller vocabulary but the model size is compa-
rable to the system A’s model size. Translation
did not benefit from source side morphological de-
composition. We also observe that the gain from
LMBR hypothesis combination (A+B+C) is mini-
mal. Unlike other language pairs, such as Arabic-
English (de Gispert et al., 2009), we have not yet
found any great advantage in multiple morpho-
logical decomposition or preprocessing analyses
of the source text. 5-gram and LMBR rescoreing
give consistent improvements. 5-gram rescoreing
improvements are very modest, probably because
the first pass 4-gram model is trained on the same
data. As noted, hypothesis combination using the
various segmentations gives consistent but modest
gains over each individual system.

Two systems were submitted to the evalu-
atation. System A+B+C achieved a mixed-cased
NIST BLEU score of 24.6, which was the top
score achieved under this measure. System A sys-
tem achieved a mixed-cased NIST BLEU score of
24.5, which was the second highest score.

4 Summary
We have successfully trained a Russian-English
system for the first time. Lessons learned include
that simple tokenisation is enough to process the
Russian side, very modest gains come from com-
bining alternative segmentations (it could also be
that the Morfessor segmentation should not be per-
formed after CoreNLP but directly on untokenised
data), and reordering between Russian and En-
glish is such that a shallow-1 grammar performs
### Table 3: Translation results, shown in lowercase NIST BLEU. Bold results correspond to submitted systems.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>newstest2012.tune</th>
<th>newstest2012.test</th>
<th>newstest2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreNLP(A)</td>
<td>33.65</td>
<td>32.36</td>
<td>25.55</td>
</tr>
<tr>
<td>+5g</td>
<td>33.67</td>
<td>32.58</td>
<td>25.63</td>
</tr>
<tr>
<td>+5g+LMBR</td>
<td><strong>33.98</strong></td>
<td><strong>32.89</strong></td>
<td><strong>25.89</strong></td>
</tr>
<tr>
<td>CoreNLP+Morfessor(B)</td>
<td>33.21</td>
<td>31.91</td>
<td>25.33</td>
</tr>
<tr>
<td>+5g</td>
<td>33.28</td>
<td>32.12</td>
<td>25.44</td>
</tr>
<tr>
<td>+5g+LMBR</td>
<td>33.58</td>
<td>32.43</td>
<td>25.78</td>
</tr>
<tr>
<td>CoreNLP+TreeTagger(C)</td>
<td>32.92</td>
<td>31.54</td>
<td>24.78</td>
</tr>
<tr>
<td>+5g</td>
<td>32.94</td>
<td>31.85</td>
<td>24.97</td>
</tr>
<tr>
<td>+5g+LMBR</td>
<td>33.12</td>
<td>32.12</td>
<td>25.05</td>
</tr>
<tr>
<td>A+B+C</td>
<td><strong>34.32</strong></td>
<td><strong>33.13</strong></td>
<td><strong>26.00</strong></td>
</tr>
</tbody>
</table>

Competitively.

Future work could include exploring alternative grammars, applying a 5-gram Kneser-Ney smoothed language model directly in first-pass decoding, and combining alternative segmentations that are more diverse from each other.

### Acknowledgments
The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-ICT-2009-4) under grant agreement number 247762. Tong Xiao was supported in part by the National Natural Science Foundation of China (Grant 61073140 and Grant 61272376) and the China Postdoctoral Science Foundation (Grant 2013M530131).

### References


Chris Dyer, Aaron Cordova, Alex Mont, and Jimmy Lin. 2008. Fast, easy, and cheap: Construction of statistical machine translation models with


