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 (Allauzen 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.[2] 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.[3] 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

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 200–205,
Sofia, Bulgaria, August 8-9, 2013

Lang	Segmentation	# Tokens	# Types
RU	CoreNLP	47.4M	1.2M
RU	Morfessor	50.0M	0.4M
RU	TreeTagger	47.4M	1.5M
EN	Cambridge	50.4M	0.7M

Table 1: Russian-English parallel corpus statistics for various segmentations.

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 *newssyscomb2009* 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.

Corpus	# Tokens
EU + NC + UN + CzEng + Yx	652.5M
Giga + CC + Wiki	654.1M
News Crawl	1594.3M
afp	874.1M
apw	1429.3M
cna + wpb	66.4M
ltw	326.5M
nyt	1744.3M
xin	425.3M
Total	7766.9M

Table 2: Statistics for English monolingual corpora.

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 $\mathbf{N}$ to be the set of non-terminals and $\mathbf{T}$ the set of terminals or words, then we can define the grammar as a set $\mathbf{R} = \{R\}$ of rules $R : N \rightarrow \langle\gamma,\alpha\rangle$ / p, where $N \in \mathbf{N}$, $\gamma, \alpha \in \{\mathbf{N} \cup \mathbf{T}\}^+$ 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 $\mathcal{L}(N, x, y)$ 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.

We are using shallow-1 hierarchical grammars (de Gispert et al., 2010) in our experiments. This model is constrained enough that the decoder can build exact search spaces, i.e. there is no pruning in search that may lead to spurious undergeneration errors.

2.6 Features and Parameter Optimisation

We use the following standard features:

- language model
- source-to-target and target-to-source translation scores
- source-to-target and target-to-source lexical scores
- target word count
- rule count
- glue rule count
- deletion rule count (each source unigram, except for OOVs, is allowed to be deleted)
- binary feature indicating whether a rule is extracted once, twice or more than twice (Bender et al., 2007)

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:

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

where **ru** are the terminals in the Russian side of a rule, **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 estimated over the data described in section 2.4. Lattices 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 translation 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 recall 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 systems trained on alternative segmentations.

2.9 Post-processing

Training data is lowercased, so we apply true-casing as post-processing. We used the *disambig* 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 available data. A Kneser-Ney smoothed 4-gram language model is also trained on the following corpora: 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 casing for the word *Bundesrechnungshof*. Other rules are post-editing rules which force some words to their upper-case forms, such as *euro* $\rightarrow$ *Euro*. Post-editing rules are developed based on high-frequency errors on the *newstest2012.tune* development set. These rules give an improvement of 0.2 mixed-cased NIST BLEU on the development set.

Finally, the output is detokenised before submission and Cyrillic characters are transliterated. We assume for human judgment purposes that it is better to have a non English word in Latin alphabet than in Cyrillic (e.g. *uprazdnyayushchie*); sometimes, transliteration can also give a correct output (e.g. *Movember*), especially in the case of proper nouns.

3 Results and Discussion

Results are reported in Table 3. We use the internationalisation switch for the NIST BLEU scoring script in order to properly lowercase the hypothesis and the reference. This introduces a slight discrepancy with official results going into the English language. The *newstest2012.test* development 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 comparable to the system A's model size. Translation did not benefit from source side morphological decomposition. We also observe that the gain from LMBR hypothesis combination (A+B+C) is minimal. Unlike other language pairs, such as Arabic-English (de Gispert et al., 2009), we have not yet found any great advantage in multiple morphological decomposition or preprocessing analyses of the source text. 5-gram and LMBR rescoring give consistent improvements. 5-gram rescoring 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 evaluation. 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 system 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 combining alternative segmentations (it could also be that the Morfessor segmentation should not be performed after CoreNLP but directly on untokenised data), and reordering between Russian and English is such that a shallow-1 grammar performs

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

Table 3: Translation results, shown in lowercase NIST BLEU. Bold results correspond to submitted systems.

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

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of CIAA*, pages 11–23.

Oliver Bender, Evgeny Matusov, Stefan Hahn, Sasa Hasan, Shahram Khadivi, and Hermann Ney. 2007. The RWTH Arabic-to-English spoken language translation system. In *Proceedings of ASRU*, pages 396–401.

Graeme Blackwood. 2010. *Lattice rescoring methods for statistical machine translation*. Ph.D. thesis, Cambridge University Engineering Department and Clare College.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of EMNLP-ACL*, pages 858–867.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Jean-Cédric Chappelier and Martin Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *Proceedings of TAPD*, pages 133–137.

David Chiang, Steve DeNeefe, and Michael Pust. 2011. Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 455–460, Portland, Oregon, USA, June. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):3.

Adrià de Gispert, Sami Virpioja, Mikko Kurimo, and William Byrne. 2009. Minimum Bayes Risk Combination of Translation Hypotheses from Alternative Morphological Decompositions. In *Proceedings of HLT/NAACL, Companion Volume: Short Papers*, pages 73–76.

Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010. Hierarchical phrase-based translation with weighted finite state transducers and shallow-n grammars. In *Computational Linguistics*.

Yonggang Deng and William Byrne. 2008. Hmm word and phrase alignment for statistical machine translation. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):494–507.

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

MapReduce. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 199–207, Columbus, Ohio, June. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August.

Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009. Hierarchical phrase-based translation with weighted finite state transducers. In *Proceedings of NAACL*, pages 433–441.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP*, volume 1, pages 181–184.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation*, volume 8.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of HLT-NAACL*, pages 169–176.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734, Honolulu, Hawaii, October. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Juan Pino, Aurelien Waite, and William Byrne. 2012. Simple and efficient model filtering in statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 98(1):5–24.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Andreas Stolcke. 2002. SRILM–An Extensible Language Modeling Toolkit. In *Proceedings of ICSLP*, volume 3, pages 901–904.

Roy W. Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proceedings of EMNLP*, pages 620–629.