NRC Russian-English Machine Translation System for WMT 2016

Chi-kiu Lo **Colin Cherry** **George Foster** **Darlene Stewart** **Rabib Islam**
Anna Kazantseva **Roland Kuhn**
National Research Council Canada
1200 Montreal Road, Ottawa, Ontario K1A 0R6, Canada
FirstName.LastName@nrc.ca

Abstract

We describe the statistical machine translation system developed at the National Research Council of Canada (NRC) for the Russian-English news translation task of the First Conference on Machine Translation (WMT 2016). Our submission is a phrase-based SMT system that tackles the morphological complexity of Russian through comprehensive use of lemmatization. The core of our lemmatization strategy is to use different views of Russian for different SMT components: word alignment and bilingual neural network language models use lemmas, while sparse features and reordering models use fully inflected forms. Some components, such as the phrase table, use both views of the source. Russian words that remain out-of-vocabulary (OOV) after lemmatization are transliterated into English using a statistical model trained on examples mined from the parallel training corpus. The NRC Russian-English MT system achieved the highest uncased BLEU and the lowest TER scores among the eight participants in WMT 2016.

1 Introduction

We present NRC's submission to the Russian-English news translation task of WMT 2016. Russian-English is a challenging language pair for statistical machine translation because Russian is a highly inflectional and free word order language. Case information is encoded by modifying the Russian words, which makes the number of word types present in the Russian side of a Russian-English parallel corpus much higher than in the English side, introducing a data sparsity problem.

Lemmatization is one of the possible solutions for handling data sparsity when translating highly inflectional languages. However, Russian is a free word order language, meaning that case information conveyed through inflection plays an important role in disambiguating the meaning of a sentence. The MT system would be unable to recover this case information if we were to blindly lemmatize all the Russian words to their root form.

Instead, we rely most heavily on lemmatization only when the missing inflections are unlikely to cause ambiguity. For example, in automatic word alignment, the missing case information should not confuse the system as competing inflections are unlikely to appear in the same sentence (El Kholy and Habash, 2012). Therefore, we build automatic word alignments with lemmatized Russian, but then restore the Russian lemmas to their inflected forms before estimating our other model parameters. The end result is a system with higher-quality word alignments, but which can still use case information to drive its translation and reordering models. Similarly, our bilingual language models have large source context windows that allow them to resolve ambiguities introduced by lemmatization, so we build these based on lemmatized versions of the source by default. These include neural network joint models (NNJMs) and lexical translation models (NNLTMs) (Devlin et al., 2014).

We have found that blind lemmatization of phrase tables is actually quite harmful to translation, but Russian morphology still causes a significant increase in the number of OOVs. Therefore, we built a fallback Russian lemma phrase table for the OOVs in the Russian input, implemented as an interpolated phrase table. For any remaining Russian OOVs, we use a semi-supervised transliteration system to translate the word orthographically. This character-level subsystem is trained

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 326–332,
Berlin, Germany, August 11-12, 2016. ©2016 Association for Computational Linguistics

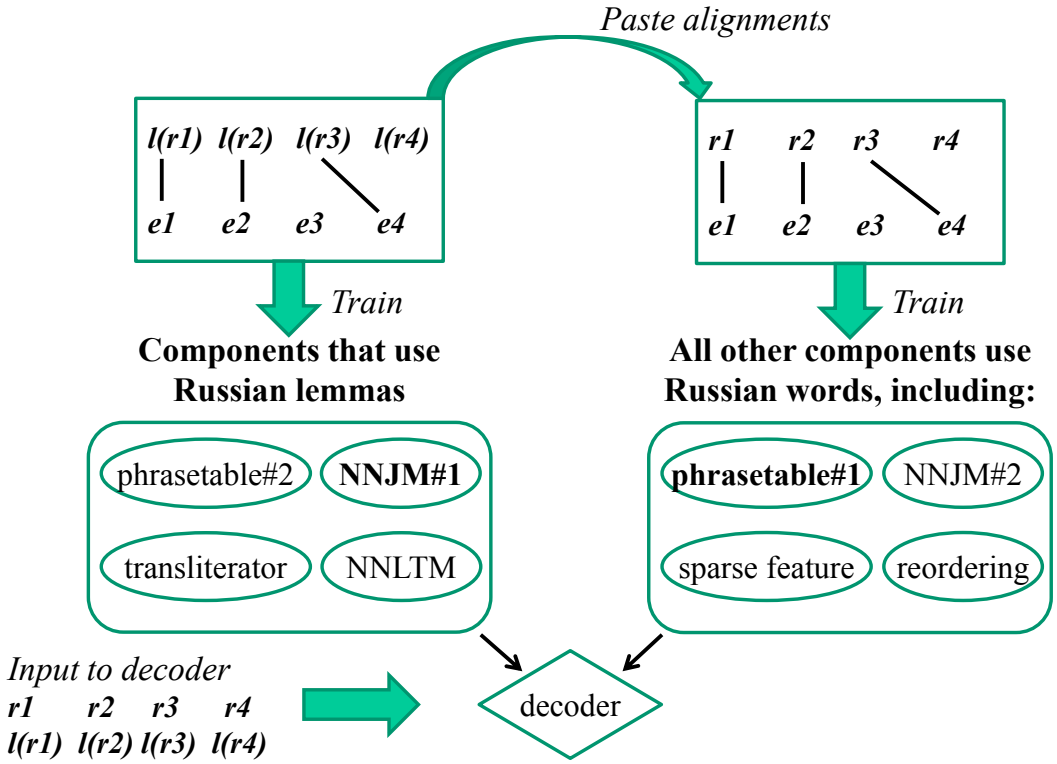

Figure 1: System diagram for the NRC Russian-English submission, highlighting our use of two different views of the Russian source. In this figure, Russian words in their inflected surface form are denoted as $r1, r2, \ldots$, while their automatically lemmatized root forms are denoted $l(r1), l(r2), \ldots$

on a transliteration corpus mined from our parallel training corpus, where the mining process is seeded by the name-pair corpora provided by the competition.

Figure 1 summarizes our lemmatization strategy. In this figure, phrasetable#1 corresponds to the phrase table given the highest weight in our interpolation (see Section 3.2), while NNJM#1 simply denotes that NNJM we found empirically to be the most informative. We did not have time to try duplicating all the models in this way; for instance, it might have been interesting to try lemma-based reordering models and an NNLTM based on Russian words rather than Russian lemmas, but we will leave this for future work.

The NRC submission achieved the highest uncased BLEU, the second highest cased BLEU and the lowest TER scores among the eight participants in the task, and ranked third out of ten systems in the human evaluation.

2 Portage - the NRC PBMT system

The core of the NRC MT system is *Portage* (Larkin et al., 2010). Portage is a conventional log-linear phrase-based SMT system. We describe the basic features of Portage in this section and the new features first tested on our Russian-English submission in the next section.

2.1 Data and preprocessing

We used all the Russian-English parallel corpora available for the constrained news translation task. They include the CommonCrawl corpus, the NewsCommentary v11 corpus, the Yandex corpus and the Wikipedia headlines corpus. We also added the WMT 12 and WMT 13 Russian-English news translation test set to the parallel training data. In total, 2.6 million parallel Russian-English sentences are used to train the translation model. For monolingual English corpora, we used the Gigaword corpus (191 million sentences) and the monolingual English corpus available for the constrained news translation task, which is a combination of the Europarl v7 corpus, the NewsCommentary v11 monolingual corpus and the NewsCrawl 2015 (206 million sentences in total). Due to resource limits, we have not used the newly released 3 billion sentence CommonCrawl monolingual English corpus. Our submitted system was tuned on the WMT 2014 test set. Both the Russian and English text in the parallel and monolingual corpora in the training/development/test cor-

pora were lower cased and tokenized.

2.2 Translation model

We obtain the word alignment by first lemmatizing the Russian side of the parallel training data using Yandex `MyStem` (Segalovich, 2003). Word alignments are built for the lemmatized Russian using IBM2, HMM and IBM4 models. The Russian is then restored to its fully inflected surface form, and phrase-pairs are extracted for each of our three alignment methods. Counts from all three alignments are then combined into a single phrase table, with a maximum phrase length of 7 tokens. Phrase pairs were filtered so that the top 30 translations for each source phrase were retained.

Our internal development experiments indicated that using lemma alignments improved the translation quality of a baseline phrase-based system by roughly 0.2 BLEU, and also benefited the perplexity of the bilingual neural language models described in Section 2.5 and 3.1.

2.3 Language models

Our system consists of three n-gram language models (LMs) and two word class language models (Stewart et al., 2014). Each is included as a distinct feature in the decoder's log-linear model.

- A 4-gram LM trained on the target side of all the WMT parallel training corpora.
- A 6-gram LM trained on the Gigaword corpus.
- A 6-gram LM trained on the WMT monolingual English training corpus.
- A 6-gram, 200-word-class coarse LM trained on a concatenation of the target side of all the WMT parallel training corpora and the WMT monolingual English training corpus.
- A 6-gram, 800-word-class coarse LM trained on the same corpus as the 200-word-class model.

Word classes are built using `mkcls` (Och, 1999).

2.4 Distortion and sparse feature models

Similar to the translation model, our hierarchical distortion model and sparse feature model are based on Russian words but are built on the lemmatized alignment. The sparse feature model consists of the standard sparse features proposed in Hopkins and May (2011) and sparse hierarchical distortion model features proposed in Cherry (2013).

2.5 Neural network joint model

We employ two neural network joint models, or NNJMs (Vaswani et al., 2013; Devlin et al., 2014). The NNJM is a feed-forward neural network language model that assumes access to a source sentence f and an aligned source index a_i, which points to the most influential source word for the translation of the target word e_i. The NNJM calculates the language modeling probability $p(e_i|e_{i-n+1}^{i-1}, f_{a_i-m}^{a_i+m})$, which accounts for the $n-1$ preceding target words, and for $2m+1$ words of source context, centered around f_{a_i}. Following Devlin et al. (2014), we use $n = 4$ and $m = 5$, resulting in 3 words of target context and 11 words of source context, effectively a 15-gram language model.

Our two models differ only in the rendering of their source strings, with one using lemmas, and the other using words. The lemma-to-word system achieved a development perplexity of 6.04, while the word-to-word system reached 6.78. Since our decoder's input is Russian words, the decoder needed to map words to lemmas before calculating lemma-based NNJM probabilities. This was done by running Yandex `MyStem` on the Russian source at test time, in order to build sentence-specific position-to-lemma mappings. For both models, the source link a_i is derived from IBM4 Russian-lemma to English-word alignments.

NNJM training data is pre-processed to limit vocabularies to 96K types for source or target inputs, and 32K types for target outputs. We build 400 deterministic word clusters for each corpus using `mkcls`. Any word not among the 96K / 32K most frequent words is replaced with its cluster. For our feed-forward network architecture, we used 192 units for source embeddings and 512 units for the single hidden layer. We train our models with mini-batch stochastic gradient descent, with a batch size of 128 words, and an initial learning rate of 0.3. We check our training objective on the development set every 20K batches, and if it fails to improve for two consecutive checks, the learning rate is halved. Training stops after 5 consecutive failed checks or after 90 checks. To enable efficient decoding, our models are self-normalized with a squared penalty on the

log partition function, weighted with $\alpha = 0.1$ (Devlin et al., 2014).

2.6 Tuning and decoding

The parameters of the log-linear model were tuned by optimizing BLEU on the development set using the batch variant of MIRA (Cherry and Foster, 2012). Decoding uses the cube-pruning algorithm of (Huang and Chiang, 2007) with a 7-word distortion limit.

2.7 Rescoring

We rescored 1000-best lists output from the decoder using a rescoring model (Och et al., 2004; Foster et al., 2009) consisting of 82 features: 27 decoder features and 55 additional rescoring features. The rescoring model was tuned using n-best MIRA. Of the rescoring features, 51 consisted of various IBM features for word- and lemma-aligned IBM1, IBM2, IBM4 and HMM models, as well as various other standard length, n-gram, and n-best features.

The final four features used NNJMs for rescoring, two Russian-word NNJM rescoring features and two Russian-lemma ones. Following Devlin et al. (2014), one NNJM feature rescored the 1000-best list using a English-to-Russian NNJM, where the roles of the source and target languages are reversed, while the other used a right-to-left and English-to-Russian NNJM, where the Russian target side is traversed in reverse order. These NNJM variants were trained and self-normalized using the same parameters as the NNJMs used for decoding described above in Section 2.5, the only difference being to swap source and target and reverse target word order as described above. During development, rescoring improved our uncased BLEU score by 0.4 on newstest2015.

2.8 Truecasing

The decoder was used to translate the lowercased, rescored output to mixed case using a target side LM and a truecase map. The 3-gram truecasing LM was trained on the target side of all the WMT parallel training data as well as the WMT monolingual English corpus described in Section 2.1. Beginning of sentence case was normalized before training the LM. In addition, casing information was transferred heuristically from the source to the target for OOVs and title/upper cased multi-word sequences. Beginning-of-sentence case was also restored. There were no OOVs because of transliteration (Section 3.3), so case for transliterated words was restored via a post-processing script. As a final step, the output was detokenized with rule-based methods.

3 New features

Our success in using Russian lemmas to improve word alignment and NNJMs to improve the overall system performance has inspired us to further develop new components to leverage these ideas. In this section, we describe the new features integrated with Portage in our submitted system: a neural network lexical translation model (NNLTM), a fallback Russian lemma phrase table, and a semi-supervised transliteration model.

3.1 Neural network lexical translation model

In addition to the NNJM feature described above, we also implemented the neural network lexical translation model (NNLTM) from (Devlin et al., 2014). The NNLTM is identical in structure to the NNJM except that it does not use target context. It is complementary to the NNJM because it accounts for all source words: for each source word f_j in the current sentence, it models $p(\bar{e}_{a_j}|f_{j-m}^{j+m})$, where $\bar{e}_{a_j}$ is the sequence of zero or more words aligned to f_j. Following Devlin et al. (2014), we set $m = 5$, and used 192 units for source embeddings and 512 units for the hidden layer.

We used a single NNLTM trained on source lemmas with source and target vocabulary sizes of 128K and 64K, and backoff to source classes as described above for the NNJM. On the target side, sequences of words $\bar{e}_{a_j}$ that were not among the most frequent 64K sequences were mapped to classes that depended on the `mkcls` class of their first word and their length, up to a maximum length of 2. For example, unknown word sequences A, A_B, and A_B_C get mapped to classes `mkcls`(A):1, `mkcls`(A):2, and `mkcls`(A):2 respectively.

Training and self-normalization details were identical to those for the NNJM. Perpexity on the development set was 10.41.

3.2 Fallback Russian lemma phrase table

To augment source coverage, we used an additional phrase table trained on source lemmas in a similar fashion to the regular phrase table described in Section 2.2. We combined the two tables statically prior to decoding, into a single ta-

ble with non-lemmatized source phrases. For a given source text and its lemmatized version, we first create an expansion phrase table with an entry for each source phrase in the text whose lemmatized form is present in the lemmatized phrase table. The target phrase and scores for the entry are obtained from the lemmatized table; that is, entries for different surface forms of the same lemma will have identical scores in the expansion table. We then linearly interpolate the regular and expansion tables, using epsilon probabilities for missing entries, and a weight of 0.9 on the regular table.[1] The combined table is used in a standard way during decoding.

3.3 Transliteration

We transliterated the lemmatized forms of all Russian words whose surface forms are out-of-vocabulary, regardless of whether their lemmatized forms occurred in either the standard or the lemmatized phrase tables. Transliterations were encoded as translation rules with multiple scored alternatives, similar to the approach found to be optimal by Durrani et al. (2014). We experimented with letting transliterations compete with translations of lemmatized forms from the phrase table when available, but found that using only the transliteration rules for OOVs resulted in slightly higher BLEU scores.

Transliterations were produced by two versions of our Portage PBMT system trained to map Cyrillic character sequences into Latin ones. Words containing more than 2 characters, all of which were either alphabetic or hyphens, and at least one of which was non-ASCII, were transliterated with a standard system; others (about 20% of OOVs) were transliterated using a backoff system.

The standard transliteration system was trained on parallel corpora consisting of the *wiki-guessed-names* and *wiki-guessed-patronymic-names* corpora,[2] with first and last names split into separate entries; and additionally on 200K transliterated word pairs mined from the parallel corpora as described below. Two character 6-gram language models were trained on all word types from the English side of the parallel corpora, and from the English Gigaword. The standard system used KN smoothing for phrase probabilities and an indicator feature on phrase pairs from the mined corpus.

The backoff system was intended to enforce a more literal style of transliteration appropriate for non-words. It was trained only on the *guessed-*names* corpora, with a phrase length limit of 3 and a restriction to monotonic translation.

We used a semi-supervised approach to mine transliterated word pairs from the parallel corpora, loosely modeled on the work of Sajjad et al. (2012). We first extracted candidate pairs from one-to-one word alignments where both words were longer than 2 characters and contained only alphabetic characters. Next we scored each candidate pair e, f using the formula $\log p(e|f) + \log p(f|e) - \log p_n(e, f)$, where $p(e|f)$ and $p(f|e)$ are probabilities from (character-wise) HMM models trained on the *guessed-*names* corpora, and $p_n(e, f) = p_n(e)p_n(f)$ is a character unigram model. Finally, we ranked all candidates by descending score and retained the top 200K.

	5 runs ave.		best run
System	dev	test	
word-aligned baseline	35.3	28.0	28.1
lemma-aligned baseline	35.3	28.2	28.3
+ lemma NNJM	36.1	28.7	28.8
+ word NNJM	36.3	28.8	28.8
+ NNLTM	36.3	28.8	28.9
+ fallback lemma table	36.8	29.1	29.2
+ transliteration	37.0	29.2	29.3
+ rescoring	–	–	29.7

Table 1: Selected results from our development experiments.

4 Development Experiments

We carried out a large number of development experiments throughout the design of this system, using the data conditions described in Section 2.1, with the WMT 2014 test set as our tuning set (dev), and the WMT 2015 test set as our test set. We monitored uncased BLEU on a system-tokenized version of the test set, reporting the average and the best of 5 random tuning replications.

Table 1 provides some selected results from these experiments and table 2 shows an example of how the different components improve the translation quality. The word baseline reflects a system with standard phrase-based features, reordering models, sparse features, monolingual language models and an uninterpolated phrase table. The

[1] We experimented with log-linear and backoff combinations, but these did not perform as well.

[2] Both corpora are provided as part of the official WMT 2016 Russia-to-English training data.

input	полиция карраты предъявила 20-летнему мужчине обвинение в отказе остановиться и опасном вождении .
reference	karratha police have charged a 20-year-old man with failing to stop and reckless driving .
word-aligned baseline	police charge man in 20-years punching карраты refusing to stop and dangerous driving .
lemma-aligned baseline	police charged карраты 20-years man indicted in refusing to stop and dangerous driving .
+ neural components	police charged карраты 20-years man charged with refusing to stop and dangerous driving .
+ OOV handling	karratha police charged a 20-year-old man accused of refusing to stop and dangerous driving .
+ rescoring	karratha police have charged a 20-year-old man accused of refusing to stop and dangerous driving .

Table 2: Example that shows significant improvements by using lemma alignments, adding neural components (i.e. 2NNJMs and NNLTM), adding OOV handling (i.e. fallback lemma table and transliteration) and rescoring.

alignment for all components in this word baseline is based on the surface form of the Russian word. We then replace the word alignment for all components with lemma alignment to form the lemma baseline. We then add the neural components, the fallback lemma table and the transliteration component. The rescoring step is only done on the best model as the final step before recasing and detokenizing.

Given such a strong lemma baseline, the biggest impact comes from the addition of the first NNJM. The next largest jump comes from the fallback Russian lemma phrase table, which also improved our OOV rate considerably. We were pleasantly surprised to see the transliteration component helping to the extent that it does. These sorts of point-wise vocabulary improvements do not always have a visible impact on BLEU. We are optimistic that its impact will be even more pronounced in the human evaluation.

5 Conclusion

We have presented the NRC submission to the WMT 2016 Russian-English news translation task. The key contributions of our system include 1) using Russian lemmas to improve word alignment while using the original Russian words to preserve case information in different models; 2) the incorporation of NNJMs and NNLTM; 3) a fallback Russian lemma phrase table for Russian OOVs and 4) a semi-supervised transliteration model built on a seed corpus mined from the standard parallel training data. Our system achieved the highest uncased BLEU, the second highest cased BLEU and the lowest TER scores among the eight participants in WMT 2016, and ranked third out of ten systems in the human evaluation.

References

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proc. 2012 Conf. of the N. American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada.

Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of NAACL HLT 2013*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380, Baltimore, Maryland, June.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an unsupervised transliteration model into statistical machine translation. In *EACL*, pages 148–153.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for english–arabic statistical machine translation. *Machine Translation*, 26(1):25–45.

George Foster, Boxing Chen, Eric Joanis, Howard Johnson, Roland Kuhn, and Samuel Larkin. 2009. PORTAGE in the NIST 2009 MT Evaluation. *Technical report, NRC-CNRC.*

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362. Association for Computational Linguistics.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proc. 45th Annual Meeting of the Assoc. for Comp. Linguistics*, pages 144–151, Prague, Czech Republic.

Samuel Larkin, Boxing Chen, George Foster, Uli Germann, Eric Joanis, J. Howard Johnson, and Roland Kuhn. 2010. Lessons from NRC's portage system at WMT 2010. In *5th Workshop on Statistical Machine Translation*.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander M Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Radev Dragomir. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL).*

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 469–477. Association for Computational Linguistics.

Ilya Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In *Proc. of MLMTA-2003*, Las Vegas, US.

Darlene Stewart, Roland Kuhn, Eric Joanis, and George Foster. 2014. Coarse "split and lump" bilingual language models for richer source information in SMT. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas (AMTA)*, volume 1, pages 28–41.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1387–1392.