PROMT Systems for WMT 2019 Shared Translation Task

Alexander Molchanov
PROMT LLC
17E Uralskaya str. building 3, 199155,
St. Petersburg, Russia
Alexander.Molchanov@promt.ru

Abstract

This paper describes the PROMT submissions for the WMT 2019 Shared News Translation Task. This year we participated in two language pairs and in three directions: English-Russian, English-German and German-English. All our submissions are MarianNMT-based neural systems. We use significantly more data compared to the last year. We also present our improved data filtering pipeline.

1 Introduction

This paper provides an overview of the PROMT submissions for the WMT 2019 Shared News Translation Task. This year we participate with neural MT systems for the second time. We participate in two language pairs and in three directions (English-Russian, English-German and German-English). We describe our data preparation pipelines, models training setups and present the results on the newstest sets.

The paper is organized as follows: Section 2 is a brief overview of the submitted systems. Section 3 describes the data preparation, preprocessing and statistics in detail. Section 4 provides a detailed description of the systems. In Section 5 we present and discuss the results. Section 6 concludes the paper.

2 Systems overview

We submitted three systems based on the `MarianNMT` (Junczys-Dowmunt et al., 2018) toolkit: English-Russian, English-German and German-English. All systems are unconstrained (we use the allowed data, private data and publicly available unconstrained data like OpenSubtitles). The English-German and German-English have the same architecture. The English-Russian system is slightly different as we use separate vocabularies.

3 Data

We use all data provided by the WMT organizers, private in-house parallel data and other publicly available data, mainly from the OPUS website (Tiedemann, 2012).

The Tatoeba sets as our validation sets and the newstest2018 is our test set. The reason why we choose the Tatoeba corpus for validation is that we aim at building general-domain (and not just news-domain) models. Besides, the Tatoeba corpus is available for many language pairs beyond the scope of the WMT Translation Task.

We select a small subset from training data and mix it with monolingual news with its back-translations for fine-tuning. This will be described in detail in Section 3.4 below.

3.1 Data filtering

There are several stages in our data filtering pipeline. The statistics for the final training data are shown in Table 1 (English-Russian) and Table 2 (English-German).

Basic filtering

This includes some simple length-based and source-target length ratio-based heuristics, removing tags, lines with low amount of alphabetic symbols etc. We also remove lines which appear to be emails or web-addresses. In addition, we remove lines with rare words from the Bookshop and the OpenSubtitles corpora (using frequency lists built on large monolingual corpora including all monolingual data from WMT, private data and Wikipedia dumps).

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 302–307
Florence, Italy, August 1-2, 2019.

Deduplication

We remove duplicate translations and keep only the most frequent translation for the source sentence if it repeats more than two times. This procedure is applied to some corpora, e.g. OpenSubtitles and MultiUN which contain a lot of various (and often incorrect) translations for common phrases. For example, the English phrase '*No.*' is encountered almost 100k times in the source side of the English-Russian OpenSubtitles corpus. It has more than 78k unique translations, second most popular among which is '*Да.*' ('*Yes.*' in Russian).

Corpus	#sent	#tokens EN	#tokens RU
MultiUN	14.9	440.6	415.1
Private data	12.4	120.1	96.2
OpenSubtitles	10.9	104.9	90.5
ParaCrawl	3.0	64.3	55.9
WikiPedia	1.0	21.2	18.7
Yandex corpus	0.6	16.8	15.4
CommonCrawl	0.4	10.3	9.5
NewsCommentary	0.3	6.2	5.9
TED Talks	0.1	2.4	2.1
Total	**43.6**	**786.8**	**709.3**

Table 1: Statistics for the filtered parallel English-Russian data in millions of sentences (#sent) and tokens.

Language detection

The algorithm is a fairly simple ensemble of three tools: `pycld2` [1], `langid` (Lui and Baldwin, 2012), `langdetect` [2].

Parallel segments filtering

We apply this step to low-quality data (basically, OpenSubtitles, CommonCrawl, ParaCrawl, Bookshop). We use `Hunalign` (Varga et al., 2005) to obtain basic sentence pair scores. We also extract about 30 additional features from sentence pairs and apply inhouse classifier to discard unparallel sentence pairs. It is a simple SVM classifier, and the features include source and target lengths in tokens, average token length in symbols, number of punctuation symbols in source and target etc. We do not use any categorical features.

Corpus	#sent	#tokens EN	#tokens DE
ParaCrawl	20.3	424.8	403.4
OpenSubtitles	10.5	97.3	91.1
Private data	9.2	101.3	94.5
DGT	3.2	72.9	55.4
Europarl	2.0	57.7	54.7
CommonCrawl	1.4	31.4	29.9
EUBookshop	1.3	28.6	27.1
Rapid	1.3	22.9	22.0
EMEA	1.2	12.0	11.5
JRC-Acquis	0.7	34.1	30.7
NewsCommentary	0.3	6.2	6.4
MultiUN	0.2	6.2	5.7
TED Talks	0.1	2.4	2.3
ECB	0.1	3.1	2.8
Total	**51.8**	**900.9**	**837.5**

Table 2: Statistics for the filtered parallel English-German data in millions of sentences (#sent) and tokens.

Data filtering using language models

As last year, we use the modified bilingual Moore-Lewis data selection algorithm (Axelrod et al., 2011). However, this time we apply it all training corpora. We use the English and Russian news 2018 corpora from statmt.org as the in-domain corpora. The idea is that the news corpora can be seen as high quality general-domain data. So using them in this scenario allows to remove some noisy outlying data.

We also substitute numbers and alphanumeric sequences with placeholders and sort the data according to language models scores. We use Levenshtein distance (set to a rather low threshold) to remove similar sentence pairs with similar scores. We regard such sentence pairs as useless (or even harmful) duplicates which can prevent our translation models from better and faster converging. We remove up to 15% of data using this procedure.

3.2 Data preprocessing

BPE

We use byte pair encoding (BPE) (Sennrich et al., 2016b) to encode our data to subword units. This year we use a different preprocessing scheme compared to the last year's systems. We noticed

[1] https://pypi.org/project/pycld2/
[2] https://pypi.org/project/langdetect/

that the BPE algorithm from the `OpenNMT` toolkit (Klein et al., 2017) gives better results compared to the default script `learn_bpe.py` from the MarianNMT toolkit. We see two reasons for that: 1) the BPE merge operations are learnt to distinguish subword units at the beginning, in the middle and at the end of the word and 2) the BPE merge operations can be learnt in case-insensitive mode (OpenNMT architecture supports features, so a feature can be used to handle case). Case-insensitive BPE model is very useful when dealing with a lot of different and sometimes noisy data (like, for example, OpenSubtitles where uppercase is often used to communicate emphasis). This is also crucial when dealing with legal and financial data where specific terms are written in title case or uppercase. News headlines are also often written in title case or uppercase.

As MarianNMT does not support features yet, we decided to perform a 'trick' similar to the one described in (Tamchyna et al., 2017): instead of using a feature we insert special tokens <C> and <U> after sequences in title case or uppercase. For example, a source sentence

World Championships 2017: Neil Black praises Scottish members of Team GB

is converted to

world <C> championships <C> 2017 : neil <C> black <C> pra@@ ises scottish <C> members of team <C> gb <U>

We do not use truecaser in our pipeline as it is redundant. All data is tokenized using the `Moses` toolkit (Koehn et al., 2007) tokenizer with aggressive tokenization, then the OpenNMT BPE-splitter is applied, after that we convert the case feature to separate tokens.

English-Russian system

Same as last year, we train the model with separate vocabularies due to the Cyrillic nature of Russian alphabet. Therefore we use separate BPE models for source and target with 35k and 45k merge operations respectively. We experimented with shared vocabulary following the procedure for the English-Russian pair described in (Sennrich et al., 2016b) but did not get improvements. This year, however, we train much smaller BPE models as we noticed that our NMT systems do not handle large vocabularies (70-90k) well and generate many OOVs in the output.

English-German and German-English systems

We train a joint BPE model for the English-German pair with 40k merge operations. We use a shared vocabulary and tie all embeddings of the translation models. The human parallel data for the German-English system is exactly the same as for the English-German system, the two systems only have different synthetic back-translated data.

3.3 Synthetic data

There are two types of additional synthetic training data described in detail below. The final size of the training data for the submitted systems is roughly 4 times the total size of the filtered data in Tables and 2.

Back-translated data

Back-translations (Sennrich et al., 2016a) are a common way to improve NMT models quality. As we aim at building general-domain models, we use data from Wikipedia dumps and news from statmt.org. We shuffle the Wikipedia data and randomly select a subset of appropriate size. The selected Wikipedia subset and the news subset are roughly equal in size. The size of the whole corpus used for back-translation is approximately equivalent to the size of human training data.

For the English-Russian pair we train a baseline Russian-English transformer model using the data prepared for the last year's WMT news task (Molchanov, 2018). For the German-English we also trained a transformer model using some data from OPUS as is: Europarl, DGT, JRC-Acquis, EMEA, ECB, NewsCommentary, TED2013, GlobalVoices. We use the Tatoeba corpus as our validation set in both cases. We use our final English-German model to obtain back-translations for the German-English model.

The trained systems were used to back-translate the 2017, 2018 news corpora from statmt.org and data selected from Wikipedia in Russian, German and English respectively.

Replicated data with unknown words

We apply the technique described in (Pinnis et al., 2017) to create a synthetic parallel corpus. The procedure includes the following steps: first, we perform word-alignment of our initial parallel training corpus using the fast-align tool (Dyer et al., 2013). Then, we randomly replace from one to

three unambiguously (one-to-one) aligned tokens in both source and target parallel sentences with the special <UNK> placeholder. The same pipeline is applied to both the initial and back-translated data. We train our models to reproduce the <UNK> placeholder in various contexts and use this feature for handling named entities described in Section 4.2 below.

3.4 Data for fine-tuning

We again apply the modified bilingual Moore-Lewis data selection algorithm. We use the news 2018 corpora as our in-domain data. We select 1M sentences from the human training data (excluding MultiUN and OpenSubtitles). We also randomly select 1M sentences from the news 2018 corpus with their back-translations. The same procedure is applied to both English-Russian and English-German pairs.

4 Systems architecture

This section describes the trained systems in detail. We train transformer (Vaswani et al., 2017) models for all submitted systems. We use the recipe available at the MarianNMT website[3]. The system configuration, hyperparameters and training steps follow those in the recipe. There are two minor differences: 1) we check the validation translation less frequently and set a higher early-stopping threshold to allow the model iterate over the training data a bit longer; 2) we do not use shared vocabulary for the English-Russian system because of the different alphabets in English and Russian as we mentioned earlier. For this reason we do not tie all embeddings and only tie the target embeddings to the output layer.

We trained two models - Model1 and Model2 - for the English-Russian pair with different seeds for almost five epochs each. The training data for the two models is slightly different: 1) we did not use the deduplication scheme described in Section 3.1 above for Model1; 2) we found about 350k English sentences in the Russian news 2018 corpus. These were removed from the synthetic data only before training Model2.

We trained single models for the English-German and German-English. Both models were trained for two epochs.

[3] https://github.com/marian-nmt/marian-examples/tree/master/wmt2017-transformer

4.1 Back-off to RBMT

We fall back to our rule-based system (RBMT) in several cases:

- if the NMT model output's language is other than expected. For example, we noticed that the English-Russian model sometimes generates English text (less than 1% of the test set sentences). The reasons for this were the 350k English sentences in Russian news 2018 corpus that we used for back-translation. We did not apply language filtering to the news-crawl corpora because they had been filtered by the WMT organisers until 2018. The English output is handled by the inhouse language detection tool.
- If the output contains recurring words or n-grams.
- If the output is much shorter or longer compared to the input sentence. We use simple rules based on source-translation length ratio to detect such cases.
- We also fall back to RBMT to translate very short strings (one or two words).

4.2 Handling named entities

We preserve several types of named entities (NEs): numbers, emails, alphanumeric sequencies etc. in the following way. First, we produce the baseline NMT translation without any processing. Then we validate the translation of NEs by comparing the system's output to the source sentence. The validation is simple: we search for the corresponding strings (numbers, emails etc.) in the system's output. If some of the NEs are not translated or are translated incorrectly, we replace the entities with the <UNK> placeholder in the source sentence and translate the sentence again allowing the decoder to generate unknown words in the output. Finally, we substitute the <UNK> placeholders in the output with their initial value. If the number of the <UNK> placeholders in the NMT system's output is not equal to the number of the placeholders in the source sentence, we fall back to the baseline NMT translation without NEs processing. We do not do any specific processing for proper names this time as they are handled much better by our current systems compared to our last year's submissions.

4.3 Models configuration

We use an ensemble of two fine-tuned models as our final translation system for the English-Russian pair.

We use a single fine-tuned model for the English-German system; the German-English system is a single baseline model.

We use the beam of size 12 and the `--normalize` parameter is set to 1.

5 Results and discussion

In this section we present the BLEU (Papineni et al., 2002) scores for our systems on two test sets and the analysis of the results.

The scores are presented in Table 4. Calculation is done using the `multi-bleu-detok.perl` script from the `Moses` toolkit.

System	newstest2018	newstest2019
English-Russian		
Model2018	27.4	24.7
Model1	30.4	27.7
Model2	32.1	29.6
Model1 fine-tuned	31.9	29.1
Model2 fine-tuned	32.5	30.4
Model1+Model2 fine-tuned	**32.9**	**30.8**
English-German		
Model (baseline)	40.0	38.1
Model fine-tuned	**40.4**	**38.4**
German-English		
Model (baseline)	**40.1**	**32.1**

Table 4: Results for different systems. The submitted systems are marked in bold. Model2018 stands for our last year's submitted system which we consider the baseline. Model1 and Model2 are described in Section 4 above.

We significantly outperform the baseline for the English-Russian pair - our last year's submission for the News Task, an ensemble of 4 models. The results for Model1 and Model2 show us that better data filtering leads to better translation quality.

Fine-tuning does not give us significant improvements in terms of BLEU. We should probably try new approaches to data selection for domain adaptation.

We should also note the lower quality of the German-English model compared to our models and other participants. We think this must be connected with the fact that the data used for training the German-English model was in fact filtered for training the English-German model (thus, we paid less attention to the English side of the data).

6 Conclusions and Future work

In this paper we have described our submissions for the WMT 2019 Shared News Translation Task. Overall we have made three submissions: English-Russian, English-German and German-English.

We have documented the methodology used to prepare the training data, system training set-ups, the pipelines for handling NEs and using RBMT.

We show competitive results in two out of three language pairs.

We plan our future research in several directions. First of all, data filtering improvement (especially when training models in both directions). Second, handling proper names translation into Russian. Finally, exploring other language pairs including the Chinese and Kazakh languages.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–362, Edinburgh, Scotland, UK.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of NAACL HLT 2013*, pages 644–648, Atlanta, USA.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine

Translation. *Computing Research Repository*, arXiv:1701.02810. Version 2.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of ACL 2012, System Demonstrations*, pages 25–30, Jeju, Republic of Korea.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 07*, pages 177–180, Stroudsburg, PA, USA.

Alexander Molchanov. 2018. PROMT Systems for WMT 2018 Shared Translation Task. In *Proceedings of the Third Conference on Machine Translation*, pages 460–464, Brussels, Belgium.

Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, USA.

Marcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In *Proceedings of the 20th International Conference of Text, Speech and Dialogue (TSD2017)*, pages 237–245, Prague, Czechia.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 2016b. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 35–40, Berlin, Germany.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Proceedings of The North American Chapter of the 342 Association for Computational Linguistics Conference (NAACL-07)*, pages 508–515, Rochester, NY, USA.

Aleš Tamchyna, Marion Weller-Di Marco and Alexander Fraser. 2017. Modeling Target-Side Inflection in Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 32–42, Copenhagen, Denmark.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005*, pages 590–596, Borovets, Bulgaria.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.