Abstract

This paper describes the neural machine translation systems of the University of Latvia, University of Zurich and University of Tartu. We participated in the WMT 2017 shared task on news translation by building systems for two language pairs: English↔German and English↔Latvian. Our systems are based on an attentional encoder-decoder, using BPE subword segmentation. We experimented with back-translating the monolingual news corpora and filtering out the best translations as additional training data, enforcing named entity translation from a dictionary of parallel named entities, penalizing over- and under-translated sentences, and combining output from multiple NMT systems with SMT. The described methods give 0.7 - 1.8 BLEU point improvements over our baseline systems.

1 Introduction

We describe the neural machine translation (NMT) systems developed by the joint team of the University of Latvia, University of Zurich and University of Tartu (C-3MA). Our systems are based on an attentional encoder-decoder (Bahdanau et al., 2015), using BPE subword segmentation for open-vocabulary translation with a fixed vocabulary (Sennrich et al., 2016a). This paper is organized as follows: In Section 2 we describe our translation software and baseline setups. Section 3 describes our contributions for improving the baseline translations. Results of our experiments are summarized in Section 4. Finally, we conclude in Section 5.

2 Baseline Systems

Our baseline systems were trained with two NMT and one statistical machine translation (SMT) framework. For English↔German we only trained NMT systems, for which we used Nematus (NT) (Sennrich et al., 2017). For English↔Latvian, apart from NT systems, we additionally trained NMT systems with Neural Monkey (NM) (Helcl and Libovický, 2017) and SMT systems with LetsMT! (LMT) (Vasiljevs et al., 2012).

In all of our NMT experiments we used a shared subword unit vocabulary (Sennrich et al., 2016b) of 35000 tokens. We clipped the gradient norm to 1.0 (Pascanu et al., 2013) and used a dropout of 0.2. Our models were trained with Adadelta (Zeiler, 2012) and after 7 days of training we performed early stopping.

For training the NT models we used a maximum sentence length of 50, word embeddings of size 512, and hidden layers of size 1000. For decoding with NT we used beam search with a beam size of 12.

For training the NM models we used a maximum sentence length of 70, word embeddings and hidden layers of size 600. For decoding with NM a greedy decoder was used. Unfortunately, at the time when we performed our experiments the beam search decoder for NM was still under development and we could not reliably use it.

3 Experimental Settings

3.1 Filtered Synthetic Training Data

Increasing the training data with synthetic back-translated corpora has proven to be useful in previous work (Sennrich et al., 2016a). The method
Table 1: Example sentences translated from Latvian into English that were filtered out from the back-translated news data.

| Source | Hypothesis | Perplexity |
|--------|------------|------------|
| šodien , 21 : 16 | Sheodiennial | 70455722055883 |
| lai izdzīvotu , nepieciešams aizpildīt ap 65 % , bet valsts apmaksā 10 % . | it is necessary to fill around 65th and the state is paid to the population . | 86070783032565 |
| potenciāli zaudētie mūža gadi ir gadi , kurus cilvēks būtu nodzīvojis līdz kādam noteiktam vecumam , ja nebūtu miris nelaimēs gadījumā , kādas slimības vai cita iemesla dēļ ( līdz 64 gadu vecumam ) . | potential annualised annuity is a year that would have survived to a particular old age if it is not dead in an accident or for another reason to be in the age of 64 years old . | 73076722553665 |
| tiekoties ar cilvēkiem Latvijā , ” veiksmes stāsts ” neesot jūtams . | ” we are talking about the people of Europe , ” he said . | 3.0285224517174 |

We used each of our NMT systems to back-translate 4.5 million sentences of the monolingual news corpora in each translation direction. First we removed any translations that contained at least one <unk> symbol. We trained a language model (LM) using CharRNN with 4 million sentences from the monolingual news corpora of the target languages, resulting in three character-level RNN language models - English, German and Latvian. We used these language models to get perplexity scores for all remaining translations. The translations were then ordered by perplexity and the best (lowest) scoring 50% were used together with the sources as sources and references respectively for the additional filtered synthetic in-domain corpus. We chose scoring sentences with an LM instead of relying on neural network weights because 1) it is fast, reliable and ready to use without having to modify both NMT frameworks, and 2) it is an unbiased approach to score sentences when compared to having the system score its output by itself.

To verify that the perplexity score resembles human judgments, we took a small subset of the development sets and asked manual evaluators to rate each translation from 1 to 5. We sorted the translations by manual evaluation scores and automatically obtained perplexities, and calculated the overlap between the better halves of each. Results from this manual evaluation in Table 2 show that the LM perplexity score is good enough to separate the worst from the best translations, even though the correlation with human judgments is low.

---

1 Multi-layer Recurrent Neural Networks (LSTM, GRU, RNN) for character-level language models in Torch https://github.com/karpathy/char-rnn
Some extreme examples of sentences translated from Latvian into English are listed in Table 1. The first one is just gibberish, the second is English, but makes little sense, the third one demonstrates unusual constructions like annualised annuity. The last two examples have a good perplexity score because they seem like good English, but when looking at the source, it is clear that in the fourth example there are some parts that are not translated.

As a result, the filtering approach brought an improvement of 1.1 - 4.9 BLEU (Papineni et al., 2002) on development sets and 1.5 - 2.8 BLEU on test sets when compared to using the full back-translated news corpora.

| En→De | De→En | En→Lv | Lv→En |
|-------|-------|-------|-------|
| 55%   | 56%   | 58%   | 56%   |

Table 2: Human judgment matches with LM perplexity for filtering on 200 random sentences from the newsdev2017 dataset.

3.2 Named Entity Forcing

For our experiments with English↔German we enforced the translation of named entities (NE) using a dictionary which we built on the training data distributed for WMT 2017.

First, we performed named entity recognition (NER) using spaCy\(^2\) for German and NLTK\(^3\) for English. The reason for using different tools is that the spaCy output for English differed largely from the German one. NLTK performed much more similarly to the German spaCy output and, thus, it was easier to find NE translation pairs. We only considered NEs of type “person”, “organisation” and “geographic location” for our dictionary.

Then we did word alignment using GIZA++ (Och and Ney, 2003) with the default grow-diag-final-and alignment symmetrization method. We created an entry in our translation dictionary for every pair of aligned (multi-word) NEs. Per entry we only kept the three most frequent translation options. Since there was still a lot of noise in the resulting dictionary, we decided to filter it automatically by removing entries that:

- started with a dash e.g. filtering out “-Munich” aligned to “Hamburg”
- were longer than 70 characters or five tokens e.g. filtering out “Parliament’s Committee on Economic and Monetary Affairs and Industrial Policy” aligned to “EU”
- differed from each other in length by more than 15 characters or two tokens e.g. filtering out “Georg” aligned to “Georg von Holtzbrinck”

When translating we made use of the alignment information given by the attention mechanism when translating with our NMT systems. We identified all NEs in the source text using the same tools as for the training data. For every source NE expression we searched for the most likely aligned translations by our systems via the attention matrix. We only considered source-translation pairs for which the attention to each other was highest in both directions.

Finally, for every such NE expression we checked whether there was a translation in our NE dictionary. If yes, we swapped the translation generated by our systems with the one in the dictionary. If not, we copied the NE expression from the source sentence to the target sentence. Since the attention is only given on the subword level, we needed to merge the subword units together before comparing the translations in the NE dictionary with the ones our systems produced. To avoid swapping too many correct translations, we defined some language-specific rules which, for example, took care of different cases in German.

We initially tested our approach on the newsstest2016 data (using our baseline system for the translation). For a qualitative perspective we looked at all of the NEs that were recognized in this text. We evaluated how many of them were changed by our algorithm and how many of these changes were positive, how many were negative and how many changed a wrong NE to another wrong NE. The results of this evaluation can be seen in Table 3. For newsstest2017 this approach gave a BLEU score improvement of 0.14 - 0.16.

3.3 Coverage Penalties

Under-translation and over-translation problems are results of lacking coverage in modern NMT systems (Tu et al., 2016). Attempts to address

---

\(^2\)Industrial-Strength Natural Language Processing in Python - https://spacy.io/
\(^3\)Natural Language Toolkit - http://www.nltk.org/
Table 3: Performance of NE enforcing on newstest2016 data. The table shows how many NEs were recognized, how many of those were changed by our algorithm and how many of the changes were positive, negative or neutral.

| System | En→De | De→En |
|--------|--------|--------|
| Values | abs rel (%) | abs rel (%) |
| # recogn. NEs | 4546 - | 4201 - |
| # changed NEs | 178 3.92 | 192 4.57 |
| neg → pos | 116 65.17 | 160 83.33 |
| pos → neg | 53 29.78 | 22 11.46 |
| neg → neg | 9 5.06 | 10 5.21 |

These issues include both changes at training time and decoding time. Coverage penalty (Wu et al., 2016) is an example of a decoding time modification aimed at the under-translation problem. We designed coverage penalty variations that affect the over-translation issue as well.

More specifically, the coverage penalty is a part of the scoring function \( s(Y,X) \) that we use to rank candidate translations in beam search:

\[
s(Y,X) = \log(P(Y|X)) + cp(X;Y)
\]

Coverage penalty from (Wu et al., 2016) is defined as follows:

\[
   cp(X;Y) = \beta * \sum_{i=1}^{\|X\|} \log(min(\sum_{j=1}^{\|Y\|} p_{i,j}, 1.0)) \tag{1}
\]

where \( |Y| \) is the index of the last target word generated on the current beam search step, \( |X| \) is the number of source words, and \( p_{i,j} \) is the attention probability of the \( j \)-th target word \( y_j \) on the \( i \)-th source word \( x_i \).

This expression penalizes the hypothesis if the sum of target word attentions on source words is below 1 (it is assumed that each target word is influenced by an attention probability mass equals to one; considering per word fertility might be a better choice), so it aims at reducing the under-translation problem. We extended equation 1 to penalize the hypothesis if the sum of target word attentions on source words not only below, but also above 1; we call it the coverage deviation penalty:

\[
   cdp(X;Y) = \beta * \sum_{i=1}^{\|X\|} \log(abs(1 - \sum_{j=1}^{\|Y\|} p_{i,j})) \tag{2}
\]

We also designed a perplexity penalty that implements the assumption that each target word should not be aligned with all source words by a little amount, but with some concrete parts of the source sentence. It penalizes the hypotheses where the target words have a high entropy of the attention distribution and called it the dispersion penalty:

\[
   dp(X;Y) = \beta * - \sum_{i=1}^{\|X\|} p_{i,|Y|} * \log(p_{i,|Y|}) \tag{3}
\]

Table 4 shows BLEU results. The dispersion
penalty with optimal weight improves BLEU considerably, with the change being statistically significant. We also tried combining different types of penalties, but got no improvements.

| β | cp | cdp | dp |
|---|----|-----|----|
| 0.2 | +0.3 | +0.0 | +0.0 |
| 0.4 | -1.0 | +0.0 | +0.0 |
| 1 | -3.0 | +0.1 | +0.2 |
| 3 | - | -0.2 | +0.5 |
| 5 | - | - | +0.5 |
| 7 | - | - | +0.7 |

Table 4: En→Lv BLEU score improvements with respect to different penalty types and values of β. Best score improvements are in bold.

### 3.4 Hybrid System Combination

For translating between English↔Latvian we used all 3 systems in each direction and obtained the attention alignments from the NMT systems. For each direction we chose one main NMT system to provide the final translation for each sentence and, judging by the attention alignment distribution, tried to automatically identify unsuccessful translations. Two main types of unsuccessful translations that we noticed were when the majority of alignments are connected to only one token (example in Figure 1) or when all tokens strongly align one-to-one, hinting that the source may not have been translated at all (example in Figure 2). In the case of an unsuccessful translation, the hybrid setup checks the attention alignment distribution from the second NMT system and outputs either the sentence of that or performs a final back-off to the SMT output. This approach gave a BLEU score improvement of 0.1 - 0.3.

### 3.5 Post-processing

In post-processing of translation output we aimed to fix the most common mistakes that NMT systems tend to make. We used the output attention alignments from the NMT systems to replace any `<unk>` tokens with the source tokens that align to them with the highest weight. Any consecutive repeating n-grams were replaced with a single n-gram. The same was applied to repeating n-grams that have a preposition between them, i.e., *victim of the victim*. This approach gave a BLEU score improvement of 0.1 - 0.2.

### 4 Results

The results of our English↔German systems are summarized in Table 5 and the results of our English↔Latvian systems - in Table 6. As mentioned in the subsections of Section 3 - each implemented modification gives a little improvement in the automated evaluation. Some modifications gave either no improvement for one or both language pairs or lead to lower automated evaluation results. These were either used for only the language pair that did show improvements on the development data or not used at all in the final setup.

#### Table 5: Experiment results for translating between English↔German. Submitted systems are in bold.

| System          | En→De Dev | Test | De→En Dev | Test |
|-----------------|-----------|------|-----------|------|
| Baseline NT     | 27.4      | 21.0 | 31.9      | 27.2 |
| +filt. synth.   | 30.7      | 22.5 | 36.8      | 28.8 |
| +NE forcing     | 30.9      | 22.7 | 36.9      | 29.0 |

#### Table 6: Experiment results for translating between English↔Latvian on development (news-dev2017) and test (news-test2017). Submitted systems are in bold.

| System                   | En→Lv Dev | Test | Lv→En Dev | Test |
|--------------------------|-----------|------|-----------|------|
| Baseline NM              | 11.9      | 11.9 | 14.6      | 12.8 |
| Baseline NT              | 12.2      | 10.8 | 13.2      | 11.6 |
| Baseline LMT             | 19.8      | 12.9 | 24.3      | 13.4 |
| +filt. synth. NM         | 16.7      | 13.5 | 15.7      | 14.3 |
| +filt. synth. NT         | 16.9      | 13.6 | 15.0      | 13.8 |
| NM+NT+LMT                | -         | 13.6 | -         | 14.3 |

### 4.1 Shared Task Results

Table 7 shows how our systems were ranked in the WMT17 shared news translation task against other submitted primary systems in the constraint track. Since the human evaluation was performed by showing evaluators only the reference translation and not the source, the human evaluation rankings are the same as BLEU, which also considers only the reference translation. One exception is the ranking for En→Lv, where an insufficient amount of evaluations were performed to cover all submitted systems, resulting in a tie for the 1st place across all but one submitted systems.
Figure 2: Attention alignment visualization of a translation, in which the all alignments are strong and mainly connected to only one-to-one. Reference translation: Keplers izmēra zvaigžņu griešanās ātrumu Plejādes zvaigznājā, hypothesis translation: Kepler measures spin rates of stars in Pleiades cluster.

Table 7: Automatic (BLEU) and human ranking of our submitted systems (C-3MA) at the WMT17 shared news translation task, only considering primary constrained systems. Human rankings are shown by clusters according to Wilcoxon signed-rank test at p-level $p \leq 0.05$, and standardized mean DA score (Ave %).

| System   | BLEU | Rank Human Cluster | Ave % |
|----------|------|-------------------|-------|
| De→En    | 6 of 7 | 6-7 of 7        | 7 of 7 |
| En→De    | 10 of 11 | 9-11 of 11      | 9 of 11 |
| En→Lv    | 11 of 12 | 1-11 of 12      | 11 of 12 |
| Lv→En    | 5 of 6  | 4-5 of 6         | 4 of 6  |

5 Conclusions

In this paper we described our submissions to the WMT17 News Translation shared task. Even though none of our systems were on the top of the list by automated evaluation, each of the implemented methods did give measurable improvements over our baseline systems. To complement the paper, we release open-source software\(^4\) and configuration examples that we used for our systems.

Acknowledgments

The authors would like to thank Tilde for providing access to the LetsMT! SMT platform and the Institute of Electronics and Computer Science for providing GPU computing resources.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. Proceedings of the International Conference on Learning Representations (ICLR).

Jindřich Helcl and Jindřich Libovický. 2017. Neural monkey: An open-source tool for sequence learning. The Prague Bulletin of Mathematical Linguistics 107(1):5–17.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. Computational linguistics 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, andWei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. ICML (3) 28:1310–1318.

Marcis Pinnis, Rihrads Krislauks, Daiga Deksne, and Toms Miks. 2017. Neural machine translation for

\(^4\)Scripts for Tartu Neural MT systems for WMT 17 - https://github.com/M4t1ss/C-3MA
morphologically rich languages with improved subword units and synthetic data. In *International Conference on Text, Speech, and Dialogue*. Springer, pages 20–27.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Lüübl, Antonio Valerio Miceli Barone, Jozef Mokry, et al. 2017. Nematus: a toolkit for neural machine translation. *EACL 2017* page 65.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. [http://www.aclweb.org/anthology/P16-1162](http://www.aclweb.org/anthology/P16-1162).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. [http://www.research.ed.ac.uk/portal/files/25478429/subword1.pdf](http://www.research.ed.ac.uk/portal/files/25478429/subword1.pdf)

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *CoRR* abs/1601.04811. [http://arxiv.org/abs/1601.04811](http://arxiv.org/abs/1601.04811).

Andrejs Vasiljevs, Raivis Skadišš, and Jörg Tiedemann. 2012. LETSMT!: A Cloud-Based Platform for Do-It-Yourself Machine Translation. In Min Zhang, editor, *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, Jeju Island, Korea, July, pages 43–48. [http://www.aclweb.org/anthology/P12-3008](http://www.aclweb.org/anthology/P12-3008).

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hitoto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144. [http://arxiv.org/abs/1609.08144](http://arxiv.org/abs/1609.08144).

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.