# JUCBNMT at WMT2018 News Translation Task: Character Based Neural Machine Translation of Finnish to English

**Sainik Kumar Mahata, Dipankar Das, Sivaji Bandyopadhyay**
Computer Science and Engineering
Jadavpur University, Kolkata, India
sainik.mahata@gmail.com,
dipankar.dipnil2005@gmail.com, sivaji_cse_ju@yahoo.com

## Abstract

In the current work, we present a description of the system submitted to WMT 2018 News Translation Shared task. The system was created to translate news text from Finnish to English. The system used a Character Based Neural Machine Translation model to accomplish the given task. The current paper documents the preprocessing steps, the description of the submitted system and the results produced using the same. Our system garnered a BLEU score of 12.9.

## 1 Introduction

Machine Translation (MT) is automated translation of one natural language to another using computer software. Translation is a tough task, not only for computers, but humans as well as it incorporates a thorough understanding of the syntax and semantics of both languages. For any MT system to return good translations, it needs good quality and sufficient amount of parallel corpus (Mahata et al., 2016, 2017).

In the modern context, MT systems can be categorized into Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). SMT has had its share in making MT very popular among the masses. It includes creating statistical models, whose input parameters are derived from the analysis of bilingual text corpora, created by professional translators (Weaver, 1955). The state-of-art for SMT is Moses Toolkit[1], created by Koehn et al. (2007), incorporates subcomponents like Language Model generation, Word Alignment and Phrase Table generation. Various works have been done in SMT (Lopez, 2008; Koehn, 2009) and it has shown good results for many language pairs.

On the other hand NMT (Bahdanau et al., 2014), though relatively new, has shown considerable improvements in the translation results when compared to SMT (Mahata et al., 2018). This includes better fluency of the output and better handling of the Out-of-Vocabulary problem. Unlike SMT, it doesn't depend on alignment and phrasal unit translations (Kalchbrenner and Blunsom, 2013). On the contrary, it uses an Encoder-Decoder approach incorporating Recurrent Neural Cells (Cho et al., 2014). As a result, when given sufficient amount of training data, it gives much more accurate results when compared to SMT (Doherty et al., 2010; Vaswani et al., 2013; Liu et al., 2014).

Further, NMT can be of two types, namely Word Level NMT and Character Level NMT. Word Level NMT, though very successful, suffers from a few disadvantages. It are unable to model rare words (Lee et al., 2016). Also, since it does not learn the morphological structure of a language it suffers when accommodating morphologically rich languages (Ling et al., 2015). We can address this issue, by training the models with huge parallel corpus, but, this in turn, produces very complex and resource consuming models that aren't feasible enough.

To combat this, we plan to use Character level NMT, so that it can learn the morphological aspects of a language and construct a word, character by character, and hence tackle the rare word occurrence problem to some extent.

In the current work, we participated in the WMT 2018 News Translation Shared Task[2] that focused on translating news text, for European language pairs. The Character Based NMT system discussed in this paper was designed to accommodate Finnish to English translations. The orga-

[1] http://www.statmt.org/moses/

[2] http://www.statmt.org/wmt18/translation-task.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 445–448
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64045

nizers provided the required parallel corpora, consisting of 3,255,303 sentence pairs, for training the translation model. The statistics of the parallel corpus is depicted in Table 1 Our model was trained on a Tesla K40 GPU, and the training took around 10 days to complete.

| | |
|---|---|
| **# sentences in Fi corpus** | 3,255,303 |
| **# sentences in En corpus** | 3,255,303 |
| **# words in Fi corpus** | 53,753,718 |
| **# words in En corpus** | 73,694,350 |
| **# word vocab size for Fi corpus** | 1,065,309 |
| **# word vocab size for En corpus** | 280,822 |
| **# chars in Fi corpus** | 427,187,612 |
| **# chars in En corpus** | 405,624,094 |
| **# char vocab size for Fi corpus** | 963 |
| **# char vocab size for En corpus** | 1,360 |

Table 1: Statistics of the Finnish-English parallel corpus provided by the organizers. ”#” depicts No. of. ”Fi” and ”En” depict Finnish and English, respectively. ”char” means character and ”vocab” means vocabulary of unique tokens.

The remainder of the paper is organized as follows. Section 2 will describe the methodology of creating the character based NMT model and will include the preprocessing steps, a brief summary of the encoder-decoder approach and the architecture of our system. This will be followed by the results and conclusion in Section 3 and 4, respectively.

## 2 Methodology

For designing the model we followed some standard preprocessing steps, which are discussed below.

### 2.1 Preprocessing

The following steps were applied to preprocess and clean the data before using it for training our character based neural machine translation model. We used the NLTK toolkit[3] for performing the steps.

- **Tokenization**: Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens. In our case, these tokens were words, punctuation marks, numbers. NLTK supports tokenization of Finnish as well as English texts.
- **Truecasing**: This refers to the process of restoring case information to badly-cased or non-cased text (Lita et al., 2003). Truecasing helps in reducing data sparsity.
- **Cleaning**: Long sentences (# of tokens $>$ 80) were removed.

### 2.2 Neural Machine Translation

Neural machine translation (NMT) is an approach to machine translation that uses neural networks to predict the likelihood of a sequence of words. The main functionality of NMT is based on the sequence to sequence (seq2seq) architecture, which is described in Section 2.2.1.

### 2.2.1 Sequence to Sequence Model

Sequence to Sequence learning is a concept in neural networks, that helps it to learn sequences. Essentially, it takes as input a sequence of tokens (characters in our case)

$$X = \{x_1, x_2, ..., x_{\mathrm{n}}\}$$

and tries to generate the target sequence as output

$$Y = \{y_1, y_2, ..., y_{\mathrm{m}}\}$$

where $\mathrm{x_i}$ and $\mathrm{y_i}$ are the input and target symbols respectively.

Sequence to Sequence architecture consists of two parts, an Encoder and a Decoder.

The encoder takes a variable length sequence as input and encodes it into a fixed length vector, which is supposed to summarize its meaning and taking into account its context as well. A Long Short Term Memory (LSTM) cell was used to achieve this. The uni-directional encoder reads the characters of the Finnish texts, as a sequence from one end to the other (left to right in our case),

$$\vec{h}_{\mathrm{t}} = \vec{f}_{\mathrm{enc}}(E_{\mathrm{x}}(x_{\mathrm{t}}), \vec{h}_{\mathrm{t\text{-}1}})$$

Here, $\mathrm{E_x}$ is the input embedding lookup table (dictionary), $\vec{f}_{\mathrm{enc}}$ is the transfer function for the Long Short Term Memory (LSTM) recurrent unit. The cell state $h$ and context vector $C$ is constructed and is passed on to the decoder.

The decoder takes as input, the context vector $C$ and the cell state $h$ from the encoder, and computes the hidden state at time t as,

$$s_{\mathrm{t}} = f_{\mathrm{dec}}(E_{\mathrm{y}}(y_{\mathrm{t\text{-}1}}), s_{\mathrm{t\text{-}1}}, c_{\mathrm{t}})$$

[3] https://www.nltk.org/

Subsequently, a parametric function $\mathrm{out}_{\mathrm{k}}$ returns the conditional probability using the next target symbol $k$.

$$(y_{\mathrm{t}} = k \mid y < t, X) = \frac{1}{Z} exp(out_{\mathrm{k}}(E_{\mathrm{y}}(y_{\mathrm{t}}{-}1), s_{\mathrm{t}}, c_{\mathrm{t}}))$$

$Z$ is the normalizing constant,

$$\sum{}_{\mathrm{j}} exp(out_{\mathrm{j}}(E_{\mathrm{y}}(y_{\mathrm{t}} - 1), s_{\mathrm{t}}, c_{\mathrm{t}}))$$

The entire model can be trained end-to-end by minimizing the log likelihood which is defined as

$$L = -\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T\mathrm{y}^{\mathrm{n}}} logp(y_{\mathrm{t}} = y_{\mathrm{t}}{}^{\mathrm{n}}, y_{\mathrm{it}}{}^{\mathrm{n}}, X^{\mathrm{n}})$$

where N is the number of sentence pairs, and $\mathrm{X}^{\mathrm{n}}$ and $\mathrm{y}_{\mathrm{t}}{}^{\mathrm{n}}$ are the input sentence and the t-th target symbol in the n-th pair respectively.

The input to the decoder was one hot tensor (embeddings at character level) of English sentences while the target data was identical, but with an offset of one time-step ahead.

### 2.3 Training

For training the model, we preprocessed the Finnish and English texts to normalize the data. Thereafter, Finnish and English characters were encoded as One-Hot vectors. The Finnish characters were considered as the input to the encoder and subsequent English characters was given as input to the decoder. A single LSTM layer was used to encode the Finnish characters. The output of the encoder was discarded and only the cell states were saved for passing on to the decoder. The cell states of the encoder and the English characters were given as input to the decoder. Lastly, a Dense layer was used to map the output of the decoder to the English characters, that were mapped with an offset of 1. The *batch size* was set to 128, *number of epochs* was set to 100, activation function was *softmax*, optimizer chosen was *rmsprop* and loss function used was *categorical cross-entropy*. Learning rate was set to 0.001. The architecture of the constructed model is shown in Figure 1.

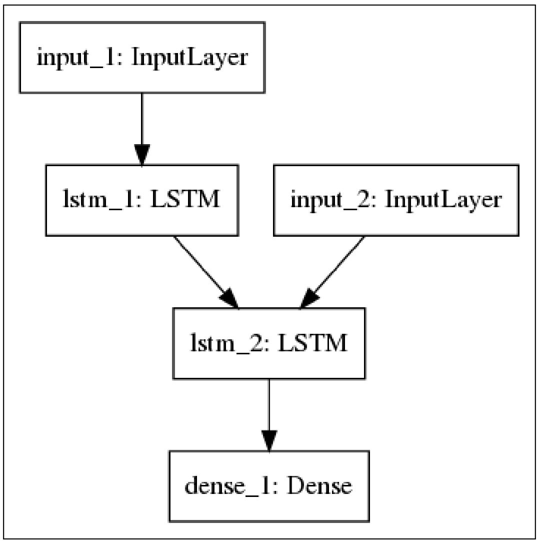

Figure 1: Architecture of the reported NMT model.

## 3 Results

Our system was a constrained system, which means that we only used data given by the organizers to train our system. The output was converted to an SGML format, the code for which was provided by the organizers. The results were submitted to `http://matrix.statmt.org/` for evaluation. The organizers calculated the BLEU score, BLEU-cased score, TER score, BEER 2.0 score, and Character TER score for our submission. As for the human ranking scores, the system fetched a standardized Average $Z$ score of -0.404 and a non-standardized Average $\%$ score of 58.9 (Bojar et al., 2018). The results of the automated and human evaluation scores are given in Table 2.

| **Metrics** | **Score** |
|---|---|
| **BLEU** | 12.9 |
| **BLEU Cased** | 12.2 |
| **TER** | 0.816 |
| **BEER 2.0** | 0.448 |
| **Character TER** | 0.770 |
| **Average** $Z$ | -0.404 |
| **Average** $\%$ | 58.9 |

Table 2: Evaluation Metrics

## 4 Conclusion

The paper presents the working of the translation system submitted to WMT 2018 News Translation shared task. We have used character based encoding for our proposed NMT system. We have used a single LSTM layer as an encoder as well as a decoder. As a future prospect, we plan to use more LSTM layers in our model. We plan to create another NMT model, which takes as input words, and not characters and subsequently use various embedding schemes to improve the translation quality.

## Acknowledgments

The reported work is supported by Media Lab Asia, MeitY, Government of India, under the Visvesvaraya PhD Scheme for Electronics & IT.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Stephen Doherty, Sharon O?Brien, and Michael Carl. 2010. Eye tracking as an mt evaluation technique. *Machine translation*, 24(1):1–13.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, volume 3, page 413.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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*, pages 177–180. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *CoRR*, abs/1610.03017.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. *CoRR*, abs/1511.04586.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3):8.

Sainik Mahata, Dipankar Das, and Santanu Pal. 2016. Wmt2016: A hybrid approach to bilingual document alignment. In *WMT*, pages 724–727.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2017. Bucc2017: A hybrid approach for identifying parallel sentences in comparable corpora. *ACL 2017*, page 56.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2018. Mtil2017: Machine translation using recurrent neural network on statistical machine translation. *Journal of Intelligent Systems*, pages 1–7.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *EMNLP*, pages 1387–1392.

Warren Weaver. 1955. Translation. *Machine translation of languages*, 14:15–23.