Byte-based Neural Machine Translation

Marta R. Costa-jussà, Carlos Escolano and **José A. R. Fonollosa**
TALP Research Center, Universitat Politècnica de Catalunya, Barcelona
marta.ruiz@upc.edu,carlos.escolano@tsc.upc.edu, jose.fonollosa@upc.edu

Abstract

This paper presents experiments comparing character-based and byte-based neural machine translation systems. The main motivation of the byte-based neural machine translation system is to build multilingual neural machine translation systems that can share the same vocabulary. We compare the performance of both systems in several language pairs and we see that the performance in test is similar for most language pairs while the training time is slightly reduced in the case of byte-based neural machine translation.

1 Introduction

Multilingual neural machine translation is raising interest in the community because it re-opens the possibility of an interlingual architecture. the main advantage of the current setting is that interlingua is not manually designed but it seems that it can be automatically extracted (Johnson et al., 2016). In addition, this multilingual environment seems to allow to build translation systems among language pairs that do not have parallel corpus available (Johnson et al., 2016), what is called "zero-shot translation".

These two motivations (interlingua and zero-shot translation) are strong enough to motivate the entire commmunity to experiment towards multilingual architectures. Recently, there have appeared works in multilingual word representations (Schwenk et al., 2017; España-Bonet et al., 2017)

Most multilingual works are at the level of words. As multilingual character research we can find (Lee et al., 2016) which goes from many-to-one languages in translation and achieves improvements for several language pairs. Previous work on character-based neural machine translation includes (Ling et al., 2015; Costa-jussà and Fonollosa, 2016), among others.

We want to explore multilingual character-based neural machine translation with a diversity of languages, including languages as Chinese. In case of using languages with different alphabets, the character dictionary can not be shared or it has to be considerably augmented. In order to keep the dictionary to the order of hundreds, we want to explore how byte-based neural machine translation behaves. In this paper, we propose to use the fully-character neural machine translation architecture (Lee et al., 2016) but using bytes instead of characters. We compare the performance of character against byte-based neural machine translation among similar languages (Catalan/Spanish and Portuguese/Brazilian) and relatively far languages (in terms of alphabet) (German/Finnish/Turkish-English).

As far as we are concerned, we are not aware of any research work in neural machine translation that has experimented with bytes. Related work can be found int the area of natural language processing. Gillick et al. (2016) propose an neural network that reads text as bytes and use this model in tasks of Part-of-Speech and Named Entity Recognition. The recent investigation of Irie et al (2017) describes the use of a byte-level convolutional layer (instead of character-level) in the neural language model (Irie et al., 2017), which is applied to low resource speech recognition.

2 Character-based Neural Machine Translation

Our system uses the architecture from (Lee et al., 2016) where a character-level neural MT model that maps the source character sequence to the target character sequence. The main difference in the encoder architecture of the standard neural

Proceedings of the First Workshop on Subword and Character Level Models in NLP, pages 154–158,
Copenhagen, Denmark, September 7, 2017.

MT model from (Bahdanau et al., 2015) is that instead of using word embeddings, the system uses character embeddings based on previous works like (Kim et al., 2015; Costa-jussà and Fonollosa, 2016). The architecture uses character embeddings include convolution layers, max pooling and highway network layers. The character embeddings from the decoder are the input of the bidirectional recurrent neural network. The main difference in the decoder architecture is that the single-layer feedforward network computes the attention score of next target character (instead of word) to be generated with every source segment representation. And afterwards, a two-layer character-level decoder takes the source context vector from the attention mechanism and predicts each target character.

3 Byte-based Neural Machine Translation

The byte-based Neural Machine Translation changes the character representation of words to the byte representation. Each sentence is represented as the concatenation of bytes that form its characters in *utf-8* encoding. No explicit vocabulay is used but we can consider the byte representation as a vocabulary of 256 positions in which every possible byte can be represented. This modifications provides the following improvements over the previously seen architecture.

- Both languages share the same representation. If a word is identical in the source and in the target language they share the same representation when converted into sequences to be fed in the network. This is an advantage over the character-based representation, which dictionary is language-dependent.
- This representation uses a limited set of tokens of size 256 independently of the language. Therefore, the system is not affected by the size of character vocabulary. Note that there are languages that have a very rich explicit morphological representation or that have a wide range of characters (e.g. Chinese). However, the byte-based decoding also produces a sequence of correct bytes in a similar way that character level translation works compared to word-based systems.
- All words are theoretically representable by the system even if they have not been previously seen in the training. This is due to the fact that every single character of word can be seen as a concatenation of bytes and the full range of possible bytes is covered by the system.

4 Experimental Framework

In this section we detail experimental corpora, architecture and parameters that we used.

4.1 Data and Preprocessing

For Catalan-Spanish, We use a large corpus extracted from ten years of the paper edition of a bilingual Catalan newspaper, El Periódico (Costa-jussà et al., 2014). The Spanish-Catalan corpus is partially available via ELDA (Evaluations and Language Resources Dis-tribution Agency) in catalog number ELRA-W0053. Development and test sets are extracted from the same corpus.

For Portuguese-Brazilian, we used the OPUS corpus (Tiedemann, 2012) which is a growing collection of translated texts from the web. In particular, for Portuguese-Brazilian the source corpus are from Ubuntu and GNOME. We extracted the parallel text from translation memories (TMX format) and from the complete text, we extracted a collection of development and test set.

Finally, we used WMT 2017 [1] corpus data for German, Finish and Turkish to English. For the three language pairs, we used all data parallel data provided in the evaluation. For German-English, we used: *europarl v.7*, *news commentary v.12*, *common crawl* and *rapid corpus of EU press releases*. For Finnish-English, we used *europarl v.8*, *wiki headlines* and *rapid corpus of EU press releases*. For Turkish-English, we used *setimes2*. The German and Finish test set is the news 2015 evaluation set, for Turkish the test set is the news 2016 evaluation set.

Preprocessing consisted in cleaning empty sentences, limiting sentences up to 50 words, tokenization and truecasing for each language using tools from Moses (Koehn et al., 2007). Table 1 shows details about the corpus statistics after preprocessing.

4.2 Parameters

Both character and byte-based systems share the same parameters. Further research may explore different parameters for the byte-based system,

[1] http://www.statmt.org/wmt17/translation-task.html

LP	L	Set	S	W	V
EsCa	Es	Train	6478618	165170227	736873
		Dev	2244	55478	12237
		Test	2244	55988	12218
	Ca	Train	6478618	178954335	713445
		Dev	2244	60130	11734
		Test	2244	60693	11691
PtBr	Pt	Train	4280310	33954616	362592
		Dev	2000	16122	4155
		Test	2000	16012	4141
	Br	Train	4280310	33600508	320972
		Dev	2000	15963	3939
		Test	2000	15663	3964
DeEn	De	Train	9659106	2 03634165	1721113
		Dev	2999	62362	12674
		Test	2169	44085	9895
	En	Train	9659106	210205446	954387
		Dev	2999	64503	9506
		Test	2169	46830	7871
FiEn	Fi	Train	2468673	3 7755811	863898
		Dev	3000	47779	16236
		Test	2870	43069	15748
	En	Train	2468673	52262051	240625
		Dev	3000	63519	9059
		Test	2870	60149	8961
TuEn	Tu	Train	200290	42 48508	158276
		Dev	1001	16954	6463
		Test	3000	54128	15898
	En	Train	299290	4713025	73906
		Dev	1001	22136	4318
		Test	3000	66394	9503

Table 1: Corpus Statistics. Number of sentences (S),words (W), vocabulary (V).

since we are adopting for the byte-based systems the character-based parameters from previous research (Lee et al., 2016).

For the embedding of the source sentence, we use set of convolutional layers which number kernels are (200-200-250-250-300-300-300-300) and their lengths are (1-2-3-4-5-6-7-8) respectively. Additionally 4 highway layers are employed. And a bidirectional LSTM layer of 512 units for encoding. The maximum souce sentence's length is 450 during training and 500 for decoding both during training and sampling.

4.3 Byte differences among language pairs

Characters may be represented by a single or several bytes. English and its close languages usually have a correspondance of character and byte (a character is represented by a single byte). Therefore, these languages are not really affected by this representation, mainly because the ASCII encoding makes possible to represent all possible characters in a single byte which results in a similar length representation in both baseline and proposed system.

However, in other languages (e.g. Turkish, Finnish...) which contain stressed characters (among other modifications), a single character in *utf-8* may be a concatenation of several bytes. For these cases, the performance of the byte-based system differs from the character-based system.

Language	**Bytes**
Spanish	1.026
Catalan	1.040
German	1.015
Finnish	1.044
Turkish	1.087
English	1.000
Portuguese	1.027
Brazilian Portuguese	1.028
Chinese	2.108

Table 2: Mean bytes for character for all the languages tested, Chinese added for comparison.

Table 2 shows the mean number of bytes for character changes for different languages. As the languages are more similiar to English the difference between bytes and characters changes. For all the languages tested in our experiments using all latin alphabet the differences are small and resultant sentence length is similar to its character counterpart.

On the other hand for languages that use a different alphabet such as Chinese we can observe how for each character more than two bytes have to be correctly generated.

5 Results

This section compares the performance of the byte-based neural machine translation system with the character-based in terms of translation quality and training time. In order to compare training times all systems have been trained in the same machine using an *NVIDIA TITAN X* with 12GB of RAM.

Table 3 shows BLEU results and number for close languages Catalan-Spanish and Portuguese-Brazilian in both directions. Comparison between character and byte-based models shows that by using a byte-base system comparable results to the ones obtained using a character-based system. In our experiments, we have observed that the byte-based system tends to converge at least a couple of hundred iterations earlier than the character-based system.

Table 4 shows BLEU results for distant languages German, Finish and Turkish into English. Comparison between character and byte-based models shows how even in distance languages similar or even equal results can be obtained us-

Language	System	BLEU
esca	char	87.12
	byte	86.91
caes	char	82.43
	byte	81.27
ptbr	char	48.54
	byte	48.28
brpt	char	46.41
	byte	47.20

Table 3: Close languages. BLEU results.

Language	System	BLEU
tren	char	5.87
	byte	4.75
deen	char	28.63
	byte	25.29
fien	char	14.75
	byte	14.75

Table 4: Distant languages. BLEU results.

ing the byte-based system. For the case of Turkish and English (and training for an equal number of iterations), results are lower than the ones obtained by the char-based system but given the reduced size of the corpus variance in the results can be caused by other factors. For Finnish-English, we achieved the same results with several hundred training iterations less. In the case of the German-English pair, results show that the char-based approach provides quite better results than the proposed byte-based system but also it is worth mentioning that when using bytes, the results converged several hundred training iterations before than when using the baseline system.

6 Conclusions

This paper shows an experimental comparison between char-based neural machine translation and byte-based neural machine translation. Variability has been found in the results for different language pairs ranging for -3.3 to $+0.8$ BLEU points. Also the main advantage of the proposed system compared to the baseline system is that it tends to require less training iterations which translates in a reduction of training time that can vary from several days to even two weeks.

Further work includes the experimentation of multilingual neural machine translation using byte-based system. In addition, we want to include in our experimentation languages with complete different alphabets such as Chinese and we will investigate if byte-based neural machine translation system proposes malformed characters. In future research, we will test different parameters such as changing the number of convolutional filters.

Acknowledgements

This work is supported by the Spanish Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional, through contract TEC2015-69266-P (MINECO/FEDER, UE) and the postdoctoral senior grant *Ramón y Cajal*.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. volume abs/1409.0473. http://arxiv.org/abs/1409.0473.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 357–361. http://anthology.aclweb.org/P16-2058.

Marta R. Costa-jussà, March Poch, José A.R. Fonollosa, Mireia Farrús, and José B. Mariño. 2014. A large Spanish-Catalna parallel corpus release for Machine Translation. *Computing and Informatics Journal* 33.

Cristina España-Bonet, Ádám Csaba Varga, Alberto Barrón-Cedeño, and Josef van Genabith. 2017. An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. *CoRR* abs/1704.05415. http://arxiv.org/abs/1704.05415.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California, pages 1296–1306.

Kazuki Irie, Pavel Golik, Ralf Schlter, and Hermann Ney. 2017. Investigations on byte-level convolutional neural networks for language modeling in low resource speech recognition page 5.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR* abs/1611.04558. http://arxiv.org/abs/1611.04558.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. *CoRR* abs/1508.06615. http://arxiv.org/abs/1508.06615.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej 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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '07, pages 177–180. http://dl.acm.org/citation.cfm?id=1557769.1557821.

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

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

Holger Schwenk, Ke Tran, Orhan Firat, and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. *CoRR* abs/1704.04154. http://arxiv.org/abs/1704.04154.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), Istanbul, Turkey, pages 2214–2218.