Transfer Learning for Digital Heritage Collections: Comparing Neural Machine Translation at the Subword-level and Character-level

Nikolay Banar[1,2], Karine Lasaracina[3], Walter Daelemans[1] and Mike Kestemont[1,2]
[1]*Computational Linguistics and Psycholinguistics Research Center, University of Antwerp, Belgium*
[2]*Antwerp Centre for Digital Humanities and Literary Criticism, University of Antwerp, Belgium*
[3]*Royal Museums of Fine Arts of Belgium, Brussels, Belgium*
{*nicolae.banari, walter.daelemans, mike.kestemont*}*@uantwerpen.be, karine.lasaracina@fine-arts-museum.be*

Keywords: Neural Machine Translation, Transfer Learning, Cultural Heritage.

Abstract: Transfer learning via pre-training has become an important strategy for the efficient application of NLP methods in domains where only limited training data is available. This paper reports on a focused case study in which we apply transfer learning in the context of neural machine translation (French–Dutch) for cultural heritage metadata (i.e. titles of artistic works). Nowadays, neural machine translation (NMT) is commonly applied at the subword level using byte-pair encoding (BPE), because word-level models struggle with rare and out-of-vocabulary words. Because unseen vocabulary is a significant issue in domain adaptation, BPE seems a better fit for transfer learning across text varieties. We discuss an experiment in which we compare a subword-level to a character-level NMT approach. We pre-trained models on a large, generic corpus and fine-tuned them in a two-stage process: first, on a domain-specific dataset extracted from Wikipedia, and then on our metadata. While our experiments show comparable performance for character-level and BPE-based models on the general dataset, we demonstrate that the character-level approach nevertheless yields major downstream performance gains during the subsequent stages of fine-tuning. We therefore conclude that character-level translation can be beneficial compared to the popular subword-level approach in the cultural heritage domain.

1 INTRODUCTION

Many cultural heritage collections are nowadays going through a phase of mass-digitization, but annotations for these datasets are still expensive and slow to obtain for smaller institutions, because they have to be provided manually by subject experts. As such, many GLAM (Galleries, Libraries, Archives, and Museums) institutions can share only small datasets with developers and researchers. Computational approaches that are effective in low-resource scenarios can therefore offer important support to cultural heritage institutions that lack the means to manually undertake large-scale cataloguing campaigns. The overall aim of this paper is to apply neural machine translation (NMT) in the context of cultural heritage metadata where only limited amounts of data are available.

NMT models require large datasets for training. However, such datasets are usually available only for the general domain. If a model is trained on one domain and applied to another one, the domain mismatch causes a significant drop in performance (Koehn and Knowles, 2017). A common remedy in such situations is the application of transfer learning. The main idea behind this concept is that the knowledge gained from one dataset can be transferred to another. In the case of neural networks, a generic network trained on a general dataset can for instance be further fine-tuned on a domain-specific dataset, which would hopefully lead to improved performance, in comparison to a system which was only trained on one of the two domains.

Nowadays, neural sequence-to-sequence networks have become a mainstream approach in machine translation. Standard NMT models operate on the word level or the subword level. The former approach requires the explicit tokenization of texts and does not handle out-of-vocabulary (OOV) words. The cultural heritage domain may contain a lot OOV words for a vocabulary obtained on the general domain, which means that such systems are not suitable for fine-tuning in our case. The subword-level approaches use byte-pair encoding (BPE) to segment sentences into tokens (Sennrich et al., 2015). BPE

is an attractive alternative to a word-level approach for our task, because it mitigates the problem of rare and OOV tokens by splitting them into more common subword units. Overall, BPE has shown strong results and is considered computationally efficient. The downside is that it requires the extensive tuning of hyperparameters for different language pairs and corpora. Additionally, the problem of finding an optimal segmentation strategy is more complicated for multilingual or zero-shot translation (Johnson et al., 2017).

Character-level models were utilized in an attempt to overcome these shortcomings. These models no longer require any explicit segmentation of the input, nor do they need the definition of language-specific vocabularies of subword items. Character-level NMT is much less sensitive to the issue of OOV words and could, in principle, more easily handle rare morphological variants of words than subword-level models (Chung et al., 2016; Lee et al., 2017). These advantages may be extremely important for the cultural heritage domain as it is very different from the general one. A character-level approach, however, also presents significant challenges compared to BPE-based models: (1) character sequences are longer and are therefore more difficult to model; (2) mapping character sequences to the more abstract level of semantics requires bigger models with a highly non-linear mapping function; and (3) such models come with a considerable increase in training/decoding time.

While the (dis)advantages mentioned above are well-known from the previous literature, this paper focuses on the issue of transfer learning in the cultural heritage domain, which has attracted less attention. We compare the performance of a BPE-based model and a character-level model, specifically in the context of further fine-tuning for the cultural heritage domain. The *downstream* flexibility of these models is crucial for the adaptation of generic background models to our domain where much more limited training data is available. Our main contribution is that we show that a BPE-based model demonstrates comparable performance to a character-level model on a large and generic corpus, but that this advantage vanishes downstream, because the BPE proves much harder to fine-tune for the cultural heritage domain. Hence, the fine-tuned character-level model shows an outspoken performance gain.

2 RELATED WORK

There has been previous work comparing character-level and subword-level NMT models. Costa-Jussa and Fonollosa applied convolutional layers to character embeddings and on top of this output they inserted highway layers (Costa-Jussa and Fonollosa, 2016). The model outperformed a NMT baseline. However, the model required the segmentation of a source sentence into words and still produced a word-level translation. Ling et al. applied a bidirectional layer of long short-term memory units (LSTM) (Hochreiter and Schmidhuber, 1997) to produce word embeddings from character embeddings (Ling et al., 2015). At the decoding side, target words were generated character by character. They showed that the model can outperform equivalent word-based NMT models. However, their approach still relied on explicit word segmentation and was acutely slow to train.

Luong and Manning developed a hybrid word-character model with a focus on solving OOV issues by using character-level information (Luong and Manning, 2016). Additionally, they implemented a fully character-level model which consisted of 4 unidirectional layers with 512 LSTM units and character-level attention. This model showed comparable results to the word-level NMT. However, the training time of the character-level model was substantially longer. Chung et al. demonstrated that a character-level decoder was able to outperform a subword-level decoder (Chung et al., 2016). This model, however, was not fully character-based, because the encoder in their architecture still resorted to BPE. Lee et al. also proposed a fully character-level NMT system (Lee et al., 2017). They designed a character-level encoding architecture that was able to efficiently model longer sequences, via the use of a convolution layer, a max-pooling layer (over the time dimension) and highway layers (Srivastava et al., 2015). Their results demonstrated that the fully character-level NMT model performed similarly to (or better than) the subword-level NMT systems.

Cherry et al., finally, demonstrated that standard character-level sequence-to-sequence models (of sufficient depth), outperformed subword-level models of comparable size (Cherry et al., 2018). Importantly for us, they also compared these models in terms of the amount of training data required. Their learning curves demonstrated that character-level models needed relatively less data to produce comparable results to subword-level models, indicating robustness in the face of limited training data. However, they did not investigate the performance of the models in the context of transfer learning for small corpora that differ from the general domain. Our research aims to fill this gap.

Table 1: Hyper-parameters of the investigated architectures. The char2char model utilizes 200 filters of width 1, 200 filters of width 2 etc.

Parameters	bpe2bpe	char2char
Source emb.	512	128
Target emb.	512	512
Conv. filters	- -	200-200-250-250 300-300-300-300
Pool stride	-	5
Highway	-	2
Encoder parameters	1-layer 512 GRUs	1-layer 512 GRUs
Decoder parameters	2-layer 1,024 GRUs	2-layer 1,024 GRUs

Table 2: Vocabulary sizes. For each language we build a BPE-based vocabulary and a character-level vocabulary.

	BPE vocab.	Char. vocab.
Vocab. size	24,400	300

3 METHODS

In this contribution, we compare a variant of the character-level NMT system (CHAR2CHAR) (Lee et al., 2017) to an established implementation of a BPE-based model (BPE2BPE), i.e. with a subword-level encoder (Sennrich et al., 2015). For both models we utilize Bahdanau Attention (Bahdanau et al., 2014) and the same architecture of decoder as in the original model. The attention score is calculated using the same input vectors as in the original implementation. We implemented both models in TensorFlow (Abadi et al., 2015). Below, we outline the details of the encoding and decoding parts of these architectures. Further information about architectures of both models is summarized in Table 1. Additionally, Table 2 provides information about vocabularies.

3.1 Sequence-to-sequence NMT

Both systems implement attentional NMT models which generally consist of the following parts: an encoder, an attention mechanism, and a decoder. Such models can be trained by minimizing the negative conditional log-likelihood.

Encoder. The encoder reads a source sentence and summarizes its meaning, typically by applying a recurrent neural network. Thus, the encoder builds a continuous representation of the input sentence.

Attention. The attention mechanism allows the model to search parts of the source sentence that are relevant for translation of each target token. It calculates the context vector for each decoding time step as a weighted sum of the source hidden states. Thus, these weights show an importance of each input token to the current target token.

Decoder. The decoder generates the output sequence based on the context vector, its previous hidden states, and the previously generated token. However, the input of the decoder may vary depending on the NMT architecture.

Table 3: Statistics of the datasets: number of sentence pairs, mean and standard deviation in sentence lengths.

Dataset	sent. pairs	mean	std
Eubookshop	2,055,656	63.89	33.08
Wiki	18,524	24.95	13.54
Museum	8,342	25.49	18.14

3.2 Subword-level NMT Model

As a representative subword-level NMT model, we utilize a recurrent sequence-to-sequence model. We briefly highlight the main properties of the encoder below.

Embedding Layer. The embedding layer maps a sequence of source tokens to a sequence of token embeddings using an embedding lookup table to create a rich representation of each token.

Recurrent Layer. Then, a bidirectional gated recurrent units (GRU) (Cho et al., 2014) layer is applied to the output of the embedding layer. A forward GRU layer reads the input sentence from left to right and a backward GRU layer reads it from right to left. Final source sentence representations are built by concatenating these layers at each timestep. We utilize GRUs because they have less parameters compared to LSTM units. Thus, the models can be easily fitted in one GPU.

Byte-Pair Encoding. This model uses BPE to mitigate the aforementioned vocabulary issues at the token level by constructing a vocabulary of the most frequently encountered word fragments to the entire word vocabulary. Hence, the length of a BPE token lies in a range from 1 character to several ones depending on the vocabulary size. BPE tokenizes the input strings by finding the longest possible match from the vocabulary in the input, or by splitting words into the longest possible fragments.

3.3 Character-level NMT Model

We implemented a variant of the character-level model (Lee et al., 2017). However, the input/output of this model can be based on other types of segmentation, such as BPE. The encoder uses one-dimensional

Table 4: Number of sentence pairs per topic extracted from Wikipedia.

	Books	Heritage	Paintings
Titles/Names	759	10,745	5,520
Descriptions	500	500	500

convolutions, alternating with max-pooling layers, in order to reduce the considerable length of the input sequence, but still efficiently capture the presence of local features. We briefly discuss the main properties of the encoder below.

Embedding Layer. As is common practice, we apply the embedding to a sequence of source characters.

Convolutions. Filter banks of one-dimensional convolutions are applied to the consecutive character embeddings in the input string (with padding). Filters have a width ranging from 1 to 8, enabling the extraction of representation of n-grams up to 8 characters. The outputs of consecutive convolutional layers are stacked and we apply the rectified linear activation.

Max Pooling. Max pooling is applied to non-overlapping segments of the output from the convolutional layer. Thus, the model produces segment embeddings that contain the most salient features of a series of characters in a particular sub-sequence of the source sentence.

Highway Layers. Highway layers (Srivastava et al., 2015) have been shown to form a crucial part of character-level models with convolutions and significantly improve the quality of the models (Kim et al., 2016). We add a stack of highway layers after the convolutional part of the encoder.

Recurrent Layer. Similarly to the BPE2BPE model, the encoder ends in a bidirectional GRU layer, which is applied to the output of the highway layers.

4 EXPERIMENTAL SETTINGS

4.1 Datasets and Preprocessing

We applied both models to a single language pair in both directions (French $\rightarrow$ Dutch and Dutch $\rightarrow$ French) in the following way: first, we pre-trained the models on a large, generic corpus and, next, we fine-tuned the pre-trained models in two different settings: (1) first, on a domain-specific dataset extracted from Wikipedia, and only then, on the actual museum metadata under scrutiny; (2) directly on the museum metadata. Additionally, we applied BPE to the CHAR2CHAR model, in order to empirically assess the influence of the segmentation procedure on this architecture.

General Corpus. As a generic background corpus, we utilize the Eubookshop (Skadiņš et al., 2014) aligned corpus (French-Dutch) to pre-train both models. We limit the length of sentence pairs to 128 characters, in order to be able to fit the CHAR2CHAR model onto a single GPU GeForce GTX TITAN X with 12 GB for a reasonable batch size (see below). Note, however, that this truncation should not impact the downstream results much, as the titles under consideration are much shorter than average sentences. We randomly selected 3,000 sentence pairs as a development set and 3,000 sentence pairs in the test set respectively.

Wikipedia Dataset. For the intermediate fine-tuning, we constructed a data set that was external to the museum metadata, but which did belong to the target domain (cultural heritage). To this end, we extracted 18,524 sentence pairs from Wikipedia[1] as domain-specific data. We found the following topics to be close to the target domain: books, heritage, paintings (see Table 4). We retrieved all pages that contained useful information and then filtered them by language while parsing. However, Wikipedia in Dutch and French is less well developed compared to the English counterpart and the extracted sentence pairs can be noisy. Many sentences, for example, are just copies of each other, and we deleted these duplicates. Additionally, descriptions can have the same structure (see Table 5) and we limit the number of them in the dataset to avoid overfitting. We use this dataset to fine-tune the pre-trained networks a first time for 2 epochs.

Museum Dataset. The museum dataset is provided by the Royal Museums of Fine Arts[2] of Belgium (RMFAB). The RMFAB maintains digital metadata about this collection via a custom-built content management system, *Fabritius*, which can be consulted online.[3] Because the RMFAB is a federal institution, significant effort is put in trying to offer the metadata in multiple languages – minimally in Dutch and French, and to a lesser degree in English). While more metadata is available and in more languages, the present paper is restricted to the titles of artworks in Dutch and French, without making a distinction between the different object categories. As the museum dataset is extremely small (see Table 3), we utilize 5-fold cross-validation to estimate the quality of the models. Thereby, in each fold, we use 3 parts to fine-tune the models for 20 epochs, 1 part to evaluate them and 1 part to control for overfitting.

[1] http://www.wikidata.org

[2] https://www.fine-arts-museum.be

[3] http://www.opac-fabritius.be

Table 5: Example of sentence pairs from the Wikipedia dataset. Descriptions are not related to Titles/Names in these examples.

Category	Language	Titles/Names	Descriptions
Books	Dutch	De beginselen van de filosofie	boek van Mac Barnett
	French	Les Principes de la philosophie	livre de Mac Barnett
Paintings	Dutch	Portret van Lorenzo Cybo	schilderij door Sophie Gengembre Anderson
	French	Portrait de Lorenzo Cybo	tableau de Sophie Gengembre Anderson
Heritage	Dutch	station Sonnenallee	archeologiemuseum in Valentano, Italië
	French	gare de Berlin Sonnenallee	musée italien

4.2 Metrics

Automated evaluation in machine translation uses established metrics to measure the quality of models. The main goal of these metrics is to replace human assessment as it is expensive and slow. Generally, the performance of automatic evaluation metrics in machine translation is measured by correlating them with human judgments. Recently, character-level metrics were observed to show the best performance among non-trainable metrics (Ma et al., 2018). Therefore, we utilize CHARACTER[4] (Wang et al., 2016) and CHRF[5] (Popović, 2015). Additionally, we apply a popular metric BLEU-4 (Papineni et al., 2002) on the character level.

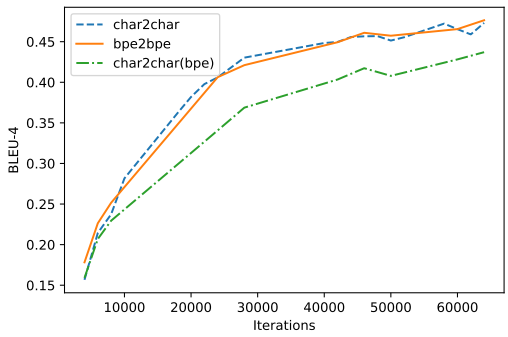

Figure 1: Example of learning curve obtained on the general dev. set. Both models with default segmentation show comparable results while training. The CHAR2CHAR model with BPE segmentation on decoder and encoder sides is worse by a large margin.

4.3 Training and Models Details

In each phase, both models were trained using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.0001 and a minibatch size of 64. The norm of the gradient is clipped with a threshold of 1. For all other parameters, we used the default TensorFlow settings. Each model is trained on a single GPU. Two epochs for the Wikipedia dataset and 20 epochs for the Museum dataset of fine-tuning are conducted for the corresponding experiments.

Our implementation of the CHAR2CHAR model slightly differs from the original implementation. Although the Highway layers significantly improve the performance of character-level language models based on convolutions, there is a saturation in performance after 2 layers (Kim et al., 2016). Accordingly, we decided to only use 2 (instead of the original 4) layers to reduce the number of parameters. For decoding, we use two-layer, unidirectional decoder with 1024 GRU units and greedy search.

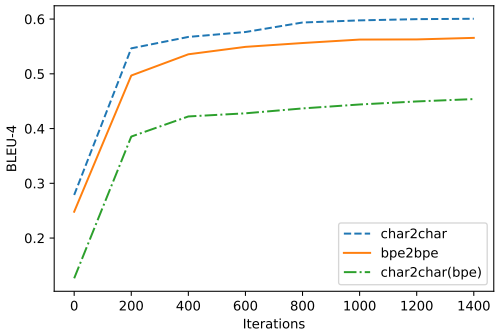

Figure 2: Example of averaged learning curve obtained in cross-validation. The CHAR2CHAR model outperforms the BPE2BPE model by a large margin.

5 RESULTS AND DISCUSSION

5.1 Quantitative Analysis

Pre-training on the General Corpus. As shown in Table 6 and Figure 1, the BPE2BPE model slightly outperforms the CHAR2CHAR model on the generic background corpus and shows at least a comparable performance for the language pair in both directions. Another interesting observation is that when changing segmentation for the CHAR2CHAR model, results substantially decrease. Additionally, we observe that the CHAR2CHAR model outperforms the BPE2BPE model when applying the pre-trained models on the new domain (see Table 6 (column gen→mus)).

[4]https://github.com/rwth-i6/CharacTER

[5]https://github.com/m-popovic/chrF

Table 6: Results of the experiments for the language pair in both directions. The arrows near the metrics correspond to a direction of improvement. Training corpora are presented on the left hand side of the horizontal arrows and test corpora are on the right hand side of them. The column label "cv" corresponds to the 5-fold fine-tuning. For these columns, the corpora on the left hand side of the horizontal arrows were used for pre-training the networks before 5-fold fine-tuning.

French→Dutch

metric	model	seg.	gen→gen	gen→mus	gen, wik→mus	gen→cv	gen, wik→cv
BLEU↑	char2char	char	**0.484**	**0.337**	**0.450**	**0.644**	**0.657**
	bpe2bpe	bpe	0.483	0.268	0.348	0.590	0.596
	char2char	bpe	0.433	0.142	0.246	0.470	0.481
CHRF↑	char2char	char	0.461	**0.293**	**0.405**	**0.585**	**0.600**
	bpe2bpe	bpe	**0.468**	0.245	0.325	0.543	0.551
	char2char	bpe	0.422	0.151	0.251	0.441	0.451
C-TER↓	char2char	char	0.543	**0.626**	**0.484**	**0.308**	**0.297**
	bpe2bpe	bpe	**0.535**	0.659	0.579	0.352	0.341
	char2char	bpe	0.575	0.784	0.660	0.462	0.450

French→Dutch

metric	model	seg.	gen→gen	gen→mus	gen, wik→mus	gen→cv	gen, wik→cv
BLEU↑	char2char	char	0.474	**0.279**	**0.410**	**0.603**	**0.614**
	bpe2bpe	bpe	**0.482**	0.248	0.332	0.567	0.578
	char2char	bpe	0.436	0.127	0.222	0.455	0.462
CHRF↑	char2char	char	0.453	**0.247**	**0.372**	**0.549**	**0.561**
	bpe2bpe	bpe	**0.468**	0.239	0.324	0.525	0.534
	char2char	bpe	0.424	0.145	0.238	0.431	0.436
C-TER↓	char2char	char	0.559	0.679	**0.537**	**0.351**	**0.343**
	bpe2bpe	bpe	**0.541**	**0.676**	0.577	0.380	0.371
	char2char	bpe	0.585	0.803	0.678	0.483	0.475

Additional Pre-training. From Table 6 (column gen, wik→mus), it can be seen that the CHAR2CHAR model benefits more from the intermediary pre-training.

Cross-validation. Downstream, however, an opposite trend can be observed and the CHAR2CHAR model outperforms the BPE2BPE model, as shown in Table 6 (column gen→cv). The gain from intermediary pre-training on the Wikipedia dataset is negligible. We conclude that the character-level model is better suited for fine-tuning at least in our task which is a typical example of a relatively small target dataset from the domain of cultural heritage.

5.2 Qualitative Error Analysis

We extracted randomly 100 sentence pairs from the Museum dataset (French→Dutch) in order to conduct a qualitative comparison between the examined models. Both models were trained using the two-stage fine-tuning scheme described above. A bilingual French-Dutch speaker evaluated the sentence pairs. The speaker was presented with the source, the target, and the outputs of the BPE2BPE and CHAR2CHAR models. We focus on differences between the models. Thus, the evaluator assigned tags to the outputs of the models where they differed, in the following categories: content words, morphology, function words and named entities. Additionally, we counted a number of fully correct sentences and named entities. Table 8 summarizes the most frequent errors that we found. Below, we highlight the main observed tendencies.

Overall Quality. We can observe that the CHAR2CHAR model slightly outperforms the BPE2BPE model in the number of the fully correct sentences. However, the number of errors that we were able to find is equal.

Named Entities. An interesting error category are named entities, such as person names, cities and countries. We found this category to be important for our task due to their frequent occurence in our dataset. We detected 42 named entities per 100 sentences. The CHAR2CHAR model correctly translated 24 entities compared to 23 entities for the BPE2BPE model. Named entities in a large majority of cases can in fact simply be copied from the input string and do not require translation. A qualitative inspection of the model outputs showed that the CHAR2CHAR model was more successful in realizing such literal copy operations. Notwithstanding that copying is not a proper translation, it may improve human perception of these

Table 7: Examples of translation from char2char and bpe2bpe models showing the main types of errors. Both models were fine-tuned on the Wikipedia and Museum datasets.

(a) Named Entities (French→Dutch)

source	La bataille de **Cassel**
target	De slag van **Kassel**
char2char	De slag bij **Cassel**
bpe2bpe	De slag bij het **Casspunt**

(b) Content words (French→Dutch)

source	Le **baiser**
target	De **kus**
char2char	De **baard**
bpe2bpe	De **bisus**

(c) Function words (French→Dutch)

source	Le départ **pour** le marché
target	Het vertrek **naar** de markt
char2char	Het vertrek **voor** de markt
bpe2bpe	Het vertrek **voor** de markt

(d) Morphology (French→Dutch)

source	Alchimiste dans son **laboratoire**
target	Alchemist in zijn **laboratorium**
char2char	Alchimist in zijn **laboratorium**
bpe2bpe	Alchist in zijn **laboratoria**

Table 8: Quantitative error analysis (when only one system mistranslates and another one predicts the correct output) and a number of correct sentences for French→Dutch. Both models were fine-tuned on the Wikipedia and Museum datasets.

Error type	**char2char**	**bpe2bpe**
Named Entities	5	6
Content words	8	9
Function words	4	0
Morphology	0	2
Total	17	17
Correct sent.	31	28

words even if they are not translated fully correct. From Table 8(a) we can observe such a case where the CHAR2CHAR model copied the location name *Cassel* from the source while the BPE2BPE model tried to translate it and failed. Both these translations are wrong, but a human can easily infer the right name from the CHAR2CHAR translation.

Content Words. According to our observation, both models often mistranslate content words. This may be related to the small sizes of the models. From Table 7(b) we can see that both models mistranslate the source content. Additionally, we noticed that the CHAR2CHAR model tends to produce spelling mistakes.

Function Words. From the example Table 7(d), we can see that both models translate the French function word *pour* as dutch *voor*, that may be right in another context. However, this is a wrong translation here.

Morphology. In the case Table 7(d) the BPE2BPE model makes a morphological mistake in the word *laboratorium*, producing the plural *laboratoria*.

6 CONCLUSION AND FUTURE WORK

We compared character-level and subword-level NMT models on the problem of transfer learning for small corpora in the domain of cultural heritage. Our experiments show that character-level NMT models are more promising at least in the context of our task. We observed a dramatic drop in performance for the CHAR2CHAR model when applying it on the subword level. The cross-validation with intermediary fine-tuning on the Wikipedia dataset slightly improved results over the cross-validation without it. However, the models tested on the Museum dataset just after intermediary fine-tuning were much better compared to the models trained just on the general corpora. Hence, we assume that a better designed dataset extracted

from Wikipedia, which was necessary with our small dataset, may help to avoid the cross-validation results in future work. Furthermore, the models may be improved in the following ways. Firstly, overall improvements may be achieved by simply increasing the sizes of the models and optimizing their hyperparameters. Secondly, a named-entity recognition system could be incorporated, because named-entities seem to be a bottleneck for both models and cause important errors. In future research we would also like to extend the research to multilingual translation. Finally, our dataset also contains images of the museum objects concerned. Thus, an interesting extension would be multi-modal translation in the domain of cultural heritage.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cherry, C., Foster, G., Bapna, A., Firat, O., and Macherey, W. (2018). Revisiting character-based neural machine translation with capacity and compression. *arXiv preprint arXiv:1808.09943*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Cho, K., and Bengio, Y. (2016). A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Costa-Jussa, M. R. and Fonollosa, J. A. (2016). Character-based neural machine translation. *arXiv preprint arXiv:1603.00810*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016). Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Lee, J., Cho, K., and Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Ling, W., Trancoso, I., Dyer, C., and Black, A. W. (2015). Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Luong, M.-T. and Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Ma, Q., Bojar, O., and Graham, Y. (2018). Results of the wmt18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Skadiņš, R., Tiedemann, J., Rozis, R., and Deksne, D. (2014). Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1850–1855.

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385.

Wang, W., Peter, J.-T., Rosendahl, H., and Ney, H. (2016). Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510.