Fully Character-Level Neural Machine Translation without Explicit Segmentation

Jason Lee[*]
ETH Zürich
jasonlee@inf.ethz.ch

Kyunghyun Cho
New York University
kyunghyun.cho@nyu.edu

Thomas Hofmann
ETH Zürich
thomas.hofmann@inf.ethz.ch

Abstract

Most existing machine translation systems operate at the level of words, relying on explicit segmentation to extract tokens. We introduce a neural machine translation (NMT) model that maps a source character sequence to a target character sequence without any segmentation. We employ a character-level convolutional network with max-pooling at the encoder to reduce the length of source representation, allowing the model to be trained at a speed comparable to subword-level models while capturing local regularities. Our character-to-character model outperforms a recently proposed baseline with a subword-level encoder on WMT'15 DE-EN and CS-EN, and gives comparable performance on FI-EN and RU-EN. We then demonstrate that it is possible to share a single character-level encoder across multiple languages by training a model on a many-to-one translation task. In this multilingual setting, the character-level encoder significantly outperforms the subword-level encoder on all the language pairs. We also observe that the quality of the multilingual character-level translation even surpasses the models trained and tuned on one language pair, namely on CS-EN, FI-EN and RU-EN.

1 Introduction

Nearly all previous work in machine translation has been at the level of words. Aside from our intuitive understanding of word as a basic unit of meaning (Jackendoff, 1992), one reason behind this is that sequences are significantly longer when represented in characters, compounding the problem of data sparsity and modeling long-range dependencies. This has driven NMT research to be almost exclusively word-level (Bahdanau et al., 2015; Sutskever et al., 2015).

Despite their remarkable success, word-level NMT models suffer from several major weaknesses. For one, they are unable to model rare, out-of-vocabulary words, making them limited in translating languages with rich morphology such as Czech, Finnish and Turkish. If one uses a large vocabulary to combat this (Jean et al., 2015), the complexity of training and decoding grows linearly with respect to the target vocabulary size, leading to a vicious cycle.

To address this, we present a fully character-level NMT model that maps a character sequence in a source language to a character sequence in a target language. We show that our model outperforms a baseline with a subword-level encoder on DE-EN and CS-EN, and achieves a comparable result on FI-EN and RU-EN. We were able to train this model at a reasonable speed by drastically reducing the length of source sentence representation using a stack of convolutional and highway layers.

One advantage of character-level models is that they are better suited for multilingual translation than their word-level counterparts which require a separate word vocabulary for each language. We verify this by training a single model to translate four languages (German, Czech, Finnish and Russian) to English. Our multilingual character-level model exceeds the subword-level baseline by a con-

[*]The majority of this work was completed while the author was visiting New York University.

siderable margin in all four language pairs, strongly indicating that a character-level model is more flexible in assigning its capacity to different language pairs. Furthermore, we also observe our multilingual character-level translation even exceeds the quality of bilingual translation in three out of four language pairs. This demonstrates excellent parameter efficiency of character-level translation in a multilingual setting. We also showcase our model's ability to handle intra-sentence code-switching, while performing language identification on the fly.

The contributions of this work are twofold: we empirically show that (1) we can train character-to-character NMT model without any explicit segmentation; and (2) we can share a single character-level encoder across multiple languages to build a multilingual translation system without increasing the model size.

2 Background: Attentional Neural Machine Translation

Neural machine translation (NMT) is a recently proposed approach to machine translation that builds a single neural network which takes as an input a source sentence $X = (x_1, \ldots, x_{T_x})$ and generates its translation $Y = (y_1, \ldots, y_{T_y})$, where x_t and $y_{t'}$ are source and target symbols (Bahdanau et al., 2015; Sutskever et al., 2015; Luong et al., 2015; Cho et al., 2014a). Attentional NMT models have three components: an *encoder*, a *decoder* and an *attention* mechanism.

Encoder Given a source sentence X, the encoder constructs a continuous representation that summarizes its meaning with a recurrent neural network (RNN). A bidirectional RNN is often implemented as proposed in (Bahdanau et al., 2015). A forward encoder reads the input sentence from left to right: $\overrightarrow{\mathbf{h}}_t = \overrightarrow{f_{\text{enc}}}\big(E_x(x_t), \overrightarrow{\mathbf{h}}_{t-1}\big)$. Similarly, a backward encoder reads it from right to left: $\overleftarrow{\mathbf{h}}_t = \overleftarrow{f_{\text{enc}}}\big(E_x(x_t), \overleftarrow{\mathbf{h}}_{t+1}\big)$, where E_x is a source embedding lookup table, and $\overrightarrow{f_{\text{enc}}}$ and $\overleftarrow{f_{\text{enc}}}$ are recurrent activation functions such as long short-term memory units (LSTMs, (Hochreiter and Schmidhuber, 1997)) or gated recurrent units (GRUs, (Cho et al., 2014b)). The encoder constructs a set of continuous source sentence representations C by concatenating the forward and backward hidden states at each timestep: $C = \big\{\mathbf{h}_1, \ldots, \mathbf{h}_{T_x}\big\}$, where $\mathbf{h}_t = \big[\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t\big]$.

Attention First introduced in (Bahdanau et al., 2015), the attention mechanism lets the decoder *attend* more to different source symbols for each target symbol. More concretely, it computes the context vector $\mathbf{c}_{t'}$ at each decoding time step t' as a weighted sum of the source hidden states: $\mathbf{c}_{t'} = \sum_{t=1}^{T_x} \alpha_{t't}\mathbf{h}_t$. Each attentional weight $\alpha_{t't}$ represents how relevant the t-th source token x_t is to the t'-th target token $y_{t'}$, and is computed as:

$$\alpha_{t't} = \frac{1}{Z}\exp\Big(\text{score}\Big(E_y(y_{t'-1}), \mathbf{s}_{t'-1}, \mathbf{h}_t\Big)\Big), \quad (1)$$

where $Z = \sum_{k=1}^{T_x} \exp\big(\text{score}(E_y(y_{t'-1}), \mathbf{s}_{t'-1}, \mathbf{h}_k)\big)$ is the normalization constant. score() is a feed-forward neural network with a single hidden layer that scores how well the source symbol x_t and the target symbol $y_{t'}$ match. E_y is the target embedding lookup table and $\mathbf{s}_{t'}$ is the target hidden state at time t'.

Decoder Given a source context vector $\mathbf{c}_{t'}$, the decoder computes its hidden state at time t' as: $\mathbf{s}_{t'} = f_{\text{dec}}\big(E_y(y_{t'-1}), \mathbf{s}_{t'-1}, \mathbf{c}_{t'}\big)$. Then, a parametric function $\text{out}_k()$ returns the conditional probability of the next target symbol being k:

$$p(y_{t'} = k | y_{<t'}, X) = \frac{1}{Z}\exp\Big(\text{out}_k\Big(E_y(y_{t'-1}), \mathbf{s}_{t'}, \mathbf{c}_{t'}\Big)\Big) \quad (2)$$

where Z is again the normalization constant: $Z = \sum_j \exp\big(\text{out}_j(E_y(y_{t'-1}), \mathbf{s}_{t'}, \mathbf{c}_{t'})\big)$.

Training The entire model can be trained end-to-end by minimizing the negative conditional log-likelihood, which is defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_y} \log p(y_t = y_t^{(n)} | y_{<t}^{(n)}, X^{(n)}),$$

where N is the number of sentence pairs, and $X^{(n)}$ and $y_t^{(n)}$ are the source sentence and the t-th target symbol in the n-th pair, respectively.

3 Fully Character-Level Translation

3.1 Why Character-Level?

The benefits of character-level translation over word-level translation are already well known. Chung et al. (2016) present three main arguments: character level models (1) do not suffer from out-of-vocabulary issues, (2) are able to model different, rare morphological variants of a word, and (3) do not require segmentation. Particularly, text segmentation is highly non-trivial for many languages and problematic even for English as word tokenizers are either manually designed or trained on a corpus using an objective function that is unrelated to the translation task at hand, which makes the overall system sub-optimal.

Here we present two additional arguments for character-level translation. First, a character-level translation system can easily be applied to a multilingual translation setting. Between European languages where the majority of alphabets overlaps, for instance, a character-level model may easily identify morphemes that are shared across different languages. A word-level model, however, will need a separate word vocabulary for each language, allowing no cross-lingual parameter sharing.

Also, by not segmenting source sentences into words, we no longer inject our knowledge of words and word boundaries into the system; instead, we encourage the model to find an internal structure of a sentence by itself and learn how a sequence of symbols can be mapped to a continuous meaning representation.

3.2 Related Work

To address these limitations associated with word-level translation, a recent line of research has investigated using sub-word information.

Costa-jussà and Fonollosa (2016) replaced the word-lookup table with convolutional and highway layers on top of character embeddings, while still segmenting source sentences into words. Target sentences were also segmented into words, and prediction was made at word-level.

Similarly, Ling et al. (2015) employed a bidirectional LSTM to compose character embeddings into word embeddings. At the target side, another LSTM takes the hidden state of the decoder and generates the target word, character by character. While this system is completely open-vocabulary, it also requires offline segmentation. Also, character-to-word and word-to-character LSTMs significantly slow down training.

Most recently, Luong and Manning (2016) proposed a hybrid scheme that consults character-level information whenever the model encounters an out-of-vocabulary word. As a baseline, they also implemented a purely character-level NMT model with 4 layers of unidirectional LSTMs with 512 cells, with attention over each character. Despite being extremely slow (approximately 3 months to train), the character-level model gave comparable performance to the word-level baseline. This shows the possibility of fully character-level translation.

Having a word-level decoder restricts the model to only being able to generate previously seen words. Sennrich et al. (2015) introduced a subword-level NMT model that is capable of open-vocabulary translation using subword-level segmentation based on the byte pair encoding (BPE) algorithm. Starting from a character vocabulary, the algorithm identifies frequent character n-grams in the training data and iteratively adds them to the vocabulary, ultimately giving a subword vocabulary which consists of words, subwords and characters. Once the segmentation rules have been learned, their model performs subword-to-subword translation (**bpe2bpe**) in the same way as word-to-word translation.

Perhaps the work that is closest to our end goal is (Chung et al., 2016), which used a subword-level encoder from (Sennrich et al., 2015) and a fully character-level decoder (**bpe2char**). Their results show that character-level decoding performs better than subword-level decoding. Motivated by this work, we aim for fully character-level translation at both sides (**char2char**).

Outside NMT, our work is based on a few existing approaches that applied convolutional networks to text, most notably in text classification (Zhang et al., 2015; Xiao and Cho, 2016). Also, we drew inspiration for our multilingual models from previous work that showed the possibility of training a single recurrent model for multiple languages in domains other than translation (Tsvetkov et al., 2016; Gillick et al., 2015).

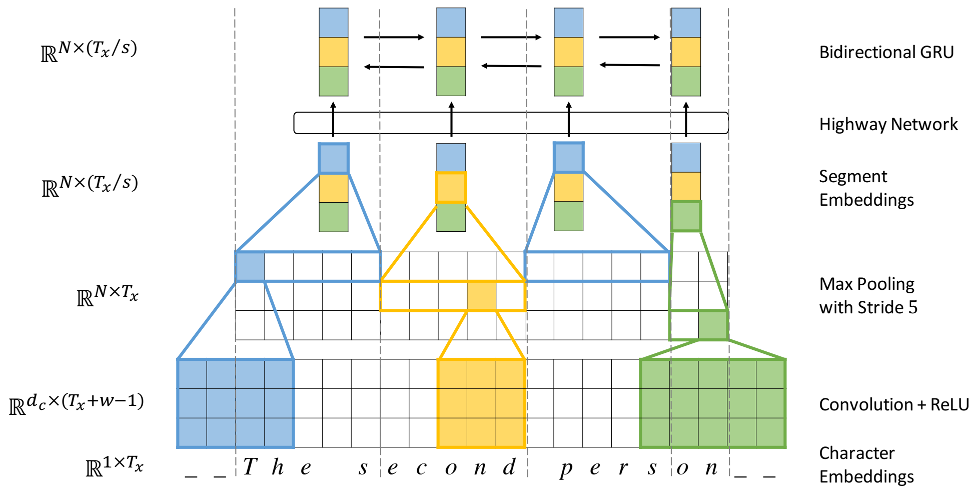

Figure 1: Encoder architecture schematics. Underscore denotes padding. A dotted vertical line delimits each segment.

3.3 Challenges

Sentences are on average 6 (DE, CS and RU) to 8 (FI) times longer when represented in characters. This poses three major challenges to achieving fully character-level translation.

(1) Training/decoding latency For the decoder, although the sequence to be generated is much longer, each character-level softmax operation costs considerably less compared to a word- or subword-level softmax. Chung et al. (2016) report that character-level decoding is only 14% slower than subword-level decoding.

On the other hand, computational complexity of the attention mechanism grows quadratically with respect to the sentence length, as it needs to attend to every source token for every target token. This makes a naive character-level approach, such as in (Luong and Manning, 2016), computationally prohibitive. Consequently, reducing the length of the source sequence is key to ensuring reasonable speed in both training and decoding.

(2) Mapping character sequence to continuous representation The arbitrary relationship between the orthography of a word and its meaning is a well-known problem in linguistics (de Saussure, 1916). Building a character-level encoder is arguably a more difficult problem, as the encoder needs to learn a highly non-linear function from a long sequence of character symbols to a meaning representation.

(3) Long range dependencies in characters A character-level encoder needs to model dependencies over longer timespans than a word-level encoder does.

4 Fully Character-Level NMT

4.1 Encoder

We design an encoder that addresses all the challenges discussed above by using convolutional and pooling layers aggressively to both (1) drastically shorten the input sentence and (2) efficiently capture local regularities. Inspired by the character-level language model from (Kim et al., 2015), our encoder first reduces the source sentence length with a series of convolutional, pooling and highway layers. The shorter representation, instead of the full character sequence, is passed through a bidirectional GRU to (3) help it resolve long term dependencies. We illustrate the proposed encoder in Figure 1 and discuss each layer in detail below.

Embedding We map the source sentence $(x_1, \ldots, x_{T_x}) \in \mathbb{R}^{1\times T_x}$ to a sequence of character embeddings $X = (\mathbf{C}(x_1), \ldots, \mathbf{C}(x_{T_x})) \in \mathbb{R}^{d_c\times T_x}$ where $\mathbf{C}$ is the character embedding lookup table: $\mathbf{C} \in \mathbb{R}^{d_c\times|\mathbf{C}|}$.

Convolution One-dimensional convolution opera-

Bilingual	bpe2char	char2char
Vocab size	24,440	300
Source emb.	512	128
Target emb.	512	512
Conv. filters		200-200-250-250 -300-300-300-300
Pool stride		5
Highway		4 layers
Encoder	1-layer 512 GRUs	
Decoder	2-layer 1024 GRUs	

Table 1: Bilingual model architectures. The char2char model uses 200 filters of width 1, 200 filters of width 2, $\cdots$ and 300 filters of width 8.

tion is then used along consecutive character embeddings. Assuming we have a single filter $\mathbf{f} \in \mathbb{R}^{d_c \times w}$ of width w, we first apply padding to the beginning and the end of X, such that the padded sentence $X' \in \mathbb{R}^{d_c \times (T_x+w-1)}$ is $w-1$ symbols longer. We then apply narrow convolution between X' and $\mathbf{f}$ such that the k-th element of the output Y_k is given as:

$$Y_k = (X' * \mathbf{f})_k = \sum_{i,j} \left(X'_{[:,k-w+1:k]} \otimes \mathbf{f}\right)_{ij}, \quad (3)$$

where $\otimes$ denotes elementwise matrix multiplication and $*$ is the convolution operation. $X'_{[:,k-w+1:k]}$ is the sliced subset of X' that contains all the rows but only w adjacent columns. The padding scheme employed above, commonly known as *half convolution*, ensures the length of the output is identical to the input's: $Y \in \mathbb{R}^{1 \times T_x}$.

We just illustrated the case of a single convolutional filter of fixed width above. In order to extract informative character patterns of different lengths, we employ a set of filters of varying widths. More concretely, we use a filter bank $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$ where $\mathbf{f}_i = \mathbb{R}^{d_c \times i \times n_i}$ is a collection of n_i filters of width i. Our model uses $m = 8$, hence extracts character n-grams up to 8 characters long. Outputs from all the filters are stacked upon each other, giving a single representation $Y \in \mathbb{R}^{N \times T_x}$, where the dimensionality of each column is given by the total number of filters $N = \sum_i^m n_i$. Finally, rectified linear activation (ReLU) is applied elementwise to this representation.

Max pooling with stride The output from the convolutional layer is first split into segments of width s, and max-pooling over time is applied to each segment. This procedure selects the most salient features to give a *segment embedding*. Each segment embedding is a summary of a particular (overlapping) segment in the source sentence; this acts as our internal linguistic unit from this layer and above: the attention mechanism, for instance, attends to each source segment instead of source character.

This shortens the source representation s-fold: $Y' \in \mathbb{R}^{N \times (T_x/s)}$. Empirically, we found using smaller s leads to better performance at increased training time. We chose $s = 5$ in our experiments.

Highway network A sequence of segment embeddings from the max pooling layer is fed into a highway network (Srivastava et al., 2015). A highway network transforms input $\mathbf{x}$ with a gating mechanism that adaptively regulates information flow:

$$\mathbf{y} = g \odot \text{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b}) + (1-g) \odot \mathbf{x},$$

where $g = \sigma((\mathbf{W}_t\mathbf{x} + \mathbf{b}_t))$. We apply this to each segment embedding individually.

Recurrent layer Finally, the output from the highway layer is given to a bidirectional GRU from §2, using each segment embedding as input.

4.2 Attention and Decoder

Similarly to the attention model in (Bahdanau et al., 2015), a single-layer feedforward network computes the attention score of next target character to be generated with every source segment representation. A two-layer character-level decoder then takes the source context vector from the attention mechanism and predicts each target character.

5 Experiment Settings

5.1 Task and Models

We evaluate the proposed character-to-character (**char2char**) translation model against subword-level baselines (**bpe2bpe** and **bpe2char**) on the WMT'15 DE→EN, CS→EN, FI→EN and

RU→EN translation tasks.[1] We compare them in two different scenarios: 1) a bilingual setting where we train a model on data from a single language pair; and 2) a multilingual setting where the task is many-to-one translation: we train a single model on data from all four language pairs. Hence, our baselines and models are:

(a) bilingual bpe2bpe: from (Firat et al., 2016a).
(b) bilingual bpe2char: from (Chung et al., 2016).
(c) bilingual char2char
(d) multilingual bpe2char
(e) multilingual char2char

We train all the models ourselves other than (a), for which we report the results from (Firat et al., 2016a). We detail the configuration of our models in Table 1 and Table 2.

5.2 Datasets and Preprocessing

We use all available parallel data on the four language pairs from WMT'15: DE-EN, CS-EN, FI-EN and RU-EN.

For the bpe2char baselines, we only use sentence pairs where the source is no longer than 50 subword symbols. For our char2char models, we only use pairs where the source sentence is no longer than 450 characters. For all the language pairs apart from FI-EN, we use newstest-2013 as a development set and newstest-2014 and newstest-2015 as test sets. For FI-EN, we use newsdev-2015 and newstest-2015 as development and test sets respectively. We tokenize each corpus using a script from Moses.[2]

When training bilingual bpe2char models, we extract 20,000 BPE operations from each of the source and target corpus using a script from (Sennrich et al., 2015). This gives a source BPE vocabulary of size 20k−24k for each language.

5.3 Training Details

Each model is trained using stochastic gradient descent and Adam (Kingma and Ba, 2014) with learning rate 0.0001 and minibatch size 64. Training continues until the BLEU score on the validation set stops improving. The norm of the gradient is clipped with a threshold of 1 (Pascanu et al., 2013). All weights are initialized from a uniform distribution $[-0.01, 0.01]$.

Each model is trained on a single pre-2016 GTX Titan X GPU with 12GB RAM.

[1] http://www.statmt.org/wmt15/translation-task.html

[2] This is unnecessary for char2char models, yet was carried out for comparison.

Multilingual	bpe2char	char2char
Vocab size	54,544	400
Source emb.	512	128
Target emb.	512	512
Conv. filters		200-200-250-250 -300-300-400-400
Pool stride		5
Highway		4 layers
Encoder	1-layer 512 GRUs	
Decoder	2-layer 1024 GRUs	

Table 2: Multilingual model architectures.

5.4 Decoding Details

As from (Chung et al., 2016), a two-layer character-level attentional decoder with 1024 GRU units is used for all our experiments. For decoding, we use beam search with length-normalization to penalize shorter hypotheses. The width of our beam is 20 for char2char models and 5 for bpe2char models.

5.5 Training Multilingual Models

Task description We train a model on a many-to-one translation task to translate a sentence in any of the four languages (German, Czech, Finnish and Russian) to English. We do *not* provide a language identifier to the encoder, but merely the sentence itself, encouraging the model to perform language identification on the fly. In addition, by not providing the language identifier, we expect the model to handle intra-sentence code-switching seamlessly.

Model architecture The multilingual char2char model uses slightly more convolutional filters than the bilingual char2char model, namely (200-200-250-250-300-300-400-400). Otherwise, the architecture remains the same as shown in Table 1. By not changing the size of the encoder and the decoder, we fix the capacity of the core translation module, and only allow the multilingual model to detect more character patterns.

Similarly, the multilingual bpe2char model has the same encoder and decoder as the bilingual bpe2char model, but a larger vocabulary. We learn 50,000 multilingual BPE operations on the multilingual corpus, resulting in 54,544 subwords. See Table 2 for the exact configuration of our multilingual models.

Data scheduling For the multilingual models, an appropriate scheduling of data from different languages is crucial to avoid overfitting to one language too soon. Following (Firat et al., 2016a; Firat et al., 2016b), each minibatch is *balanced*, in that the proportion of each language pair in a single minibatch corresponds to that of the full corpus. With this minibatch scheme, roughly the same number of updates is required to make one full pass over the entire training corpus of each language pair. Minibatches from all language pairs are combined and presented to the model as a single minibatch. See Table 3 for the minibatch size for each language pair.

	DE-EN	CS-EN	FI-EN	RU-EN
corpus size	4.5m	12.1m	1.9m	2.3m
minibatch size	14	37	6	7

Table 3: The minibatch size of each language (second row) is proportionate to the number of sentence pairs in each corpus (first row).

Treatment of Cyrillic To facilitate cross-lingual parameter sharing, we convert every Cyrillic character in the Russian source corpus to Latin alphabet according to ISO-9. Table 4 shows an example of how this conversion may help the multilingual models identify lexemes that are shared across multiple languages.

	school	schools
CS	škola	školy
RU	школа	школы
RU (ISO-9)	škola	školy

Table 4: Czech and Russian words for *school* and *schools*, alongside the conversion of Russian characters into Latin.

Multilingual BPE For the multilingual bpe2char model, multilingual BPE segmentation rules are extracted from a large dataset containing training source corpora of all the language pairs. To ensure the BPE rules are not biased towards one language, larger datasets such as Czech and German corpora are trimmed such that every corpus contains an approximately equal number of characters.

6 Quantitative Analysis

In this section, we first establish our main hypotheses for introducing character-level and multilingual models, and investigate whether our observations support or disagree with our hypotheses. From our empirical results, we want to verify: (1) if fully character-level translation outperforms subword-level translation, (2) in which setting and to what extent is multilingual translation beneficial and (3) if multilingual, character-level translation achieves superior performance to other models. We discuss each hypothesis in detail below.

(1) Character- vs. subword-level In a bilingual setting, the char2char model clearly outperforms both subword-level baselines on DE-EN (Table 5 (a-c)) and CS-EN (Table 5 (f-h)). On the other two language pairs, it exceeds the bpe2bpe model and achieves similar performance with the bpe2char baseline (Table 5 (k-m) and (p-r)). Overall, we observe that character-level translation on a single language pair performs as well as, or better than, subword-level translation.

Meanwhile, in a multilingual setting, the character-level encoder significantly surpasses the subword-level encoder consistently in all the language pairs (Table 5 (d-e), (i-j), (n-o) and (s-t)). From this, we conclude that translating at the level of characters allows the model to discover shared constructs between languages more effectively. This also demonstrates that the character-level model is more flexible in assigning model capacity to different language pairs.

(2) Multilingual vs. bilingual At the level of characters, we note that multilingual translation is indeed strongly beneficial. On the test sets, the multilingual character-level model outperforms the single-pair character-level model by 2.64 BLEU in FI-EN (Table 5 (m, o)) and 0.93 BLEU in CS-EN (Table 5 (h, j)), while achieving comparable results on DE-EN and RU-EN.

		Setting	Src	Trg	Dev	Test1	Test2
DE-EN	(a)*	bi	bpe	bpe	24.13		24.00
	(b)	bi	bpe	char	25.64	24.59	25.27
	(c)	bi	char	char	**26.32**	**25.75**	**26.01**
	(d)	multi	bpe	char	24.92	24.54	25.23
	(e)	multi	char	char	25.67	25.13	25.79
CS-EN	(f)*	bi	bpe	bpe	21.24		20.32
	(g)	bi	bpe	char	22.95	23.78	22.40
	(h)	bi	char	char	23.38	24.08	22.46
	(i)	multi	bpe	char	23.27	24.27	22.42
	(j)	multi	char	char	**24.09**	**25.01**	**23.24**
FI-EN	(k)*	bi	bpe	bpe	13.15		12.24
	(l)	bi	bpe	char	14.54		13.98
	(m)	bi	char	char	14.18		13.10
	(n)	multi	bpe	char	14.70		14.40
	(o)	multi	char	char	**15.96**		**15.74**
RU-EN	(p)*	bi	bpe	bpe	21.04		22.44
	(q)	bi	bpe	char	21.68	26.21	22.83
	(r)	bi	char	char	21.75	**26.80**	22.73
	(s)	multi	bpe	char	21.75	26.31	22.81
	(t)	multi	char	char	**22.20**	26.33	**23.33**

Table 5: BLEU scores of five different models on four language pairs. For each test or development set, the best performing model is shown in bold. $(*)$ results are taken from (Firat et al., 2016a).

At the level of subwords, on the other hand, we do not observe the same degree of performance benefit from multilingual translation. Also, the multilingual bpe2char model requires much more updates to reach the performance of the bilingual bpe2char model (see Figure 2). This suggests that learning useful subword segmentation across languages is difficult.

(3) Multilingual char2char vs. others The multilingual char2char model is the best performer in CS-EN, FI-EN and RU-EN (Table 5 (j, o, t)), and is the runner-up in DE-EN (Table 5 (e)). The fact that the multilingual char2char model outperforms the single-pair models goes to show the parameter efficiency of character-level translation: instead of training N separate models for N language pairs, it is possible to get better performance with one multilingual character-level model.

7 Qualitative Analysis

In Table 6, we demonstrate our character-level model's robustness in four translation scenarios that conventional NMT systems are known to suffer in. We also showcase our model's ability to seamlessly handle intra-sentence *code-switching*, or mixed utterances from two or more languages. We compare sample translations from the character-level model with those from the subword-level model, which already sidesteps some of the issues associated with word-level translation.

With real-world text containing typos and spelling mistakes, the quality of word-based translation would severely drop, as every non-canonical form of a word cannot be represented. On the other hand, a character-level model has a much better chance recovering the original word or sentence. Indeed, our char2char model is robust against a few spelling mistakes (Table 6 (a)).

Given a long, rare word such as "Siebentausendzweihundertvierundfünfzig" (seven thousand two hundred fifty four) in Table 6 (b), the subword-level model incorrectly segments "Siebentausend" into (Sieb, ent, aus, end), instead of (Sieben, tausend); this results in an inaccurate translation. The character-level model performs better on these long, concatenative words with ambiguous segmentation.

(a) Spelling mistakes

DE ori	Warum sollten wir nicht Freunde sei ?
DE src	Warum solltne wir nich Freunde sei ?
EN ref	Why should not we be friends ?
bpe2char	Why are we to be friends ?
char2char	Why should we not be friends ?

(b) Rare words

DE src	Siebentausendzweihundertvierundfünfzig .
EN ref	Seven thousand two hundred fifty four .
bpe2char	Fifty-five Decline of the Seventy .
char2char	Seven thousand hundred thousand fifties .

(c) Morphology

DE src	Die Zufahrtsstraßen wurden gesperrt , wodurch sich laut CNN lange Rückstaus bildeten .
EN ref	The access roads were blocked off , which , according to CNN , caused long tailbacks .
bpe2char	The access roads were locked , which , according to CNN , was long back .
char2char	The access roads were blocked , which looked long backwards , according to CNN .

(d) Nonce words

DE src	Der Test ist nun über , aber ich habe keine gute Note . Es ist wie eine Verschlimmbesserung .
EN ref	The test is now over , but i don't have any good grade . it is like a worsened improvement .
bpe2char	The test is now over , but i do not have a good note .
char2char	The test is now , but i have no good note , it is like a worsening improvement .

(e) Multilingual

Multi src	Die Bewegung wird auch von dva bývalí amerikanischen ministři dopravy unterstützt , которые в 2011 году призвали конгресс двигаться в направлении meilenbasierter Abrechnung .
EN ref	The movement is also bolstered by two former U.S. Transportation secretaries , who in a 2011 report urged Congress to move in the pay-per-mile direction .
bpe2char	The movement is also supported by two former American ministers of transport , which in 2011 called for Congress to move in the direction of meeinbased reckoning .
char2char	The movement is also supported by two former American ministers of transport , which in 2011 called for Congress to move towards miles-based accounting .

Table 6: Sample translations. For each example, we show the source sentence as *src*, the human translation as *ref*, and the translations from the subword-level baseline and our character-level model as *bpe2char* and *char2char*, respectively. For (a), the original, uncorrupted source sentence is also shown (*ori*). The source sentence in (e) contains words in German (in green), Czech (in yellow) and Russian (in blue). The translations in (a-d) are from the bilingual models, whereas those in (e) are from the multilingual models.

Handling morphological inflections should also be relatively easy for a character-level model. We observe that this is indeed the case, as our char2char model correctly understands "gesperrt", a past participle form of "sperren" (to block) (Table 6 (c)).

Nonce words are terms coined for a single use. They are not actual words but are constructed in a way that humans can intuitively guess what they mean, such as *workoliday* and *friyay*. We construct a few DE-EN sentence pairs that contain German nonce words (one example shown in Table 6 (d)), and observe that the character-level model can indeed detect salient character patterns and arrive at a correct translation.

Finally, we evaluate our multilingual models' capacity to perform intra-sentence code-switching, by giving them as input mixed sentences from multiple languages. We discover that when given sentences with high degree of language intermixing, as in Table 6 (e) or Table 12 (k) in the Appendix, the multilingual bpe2char model fails to seamlessly handle alternation of languages. Overall, however, both multilingual models generate reasonable translations. This is possible because we did not provide a language identifier when training our multilingual models; as a result, they learned to understand a multilingual sentence and translate it into a coherent English sentence.

We show more examples for each category in the Appendix.

Training and decoding speed On a single Titan X GPU, we observe that our char2char models are approximately 50% slower to train than our bpe2char baselines when the same batch size was used. Our bilingual character-level models can be trained in roughly two weeks.

We further note that the bilingual bpe2char model can translate 3,000 sentences in 66.63 minutes while the bilingual char2char model requires 71.71 minutes (online, not in batch). See Table 7 for the exact details.

	Model	Time to execute 1k updates (s)	Batch size	Time to decode 3k sentences (m)
FI-EN	bpe2char	2461.72	128	66.63
	char2char	2371.93	64	71.71
Multi	bpe2char	1646.37	64	68.99
	char2char	2514.23	64	72.33

Table 7: Speed comparison. The second column shows the time taken to execute 1,000 training updates. The model makes each update after having seen one minibatch.

Further observations We also note that the multilingual models are less prone to overfitting than the bilingual models. This is particularly visible for low-resource language pairs such as FI-EN. Figure 2 shows the evolution of the FI-EN validation BLEU scores where the bilingual models overfit rapidly but the multilingual models seem to regularize learning by training simultaneously on other language pairs.

8 Conclusion

We propose a fully character-level NMT model that accepts a sequence of characters in the source language and outputs a sequence of characters in the target language. What is remarkable about this model is the absence of explicitly hard-coded knowledge of words and their boundaries, and that the model learns these concepts from a translation task alone.

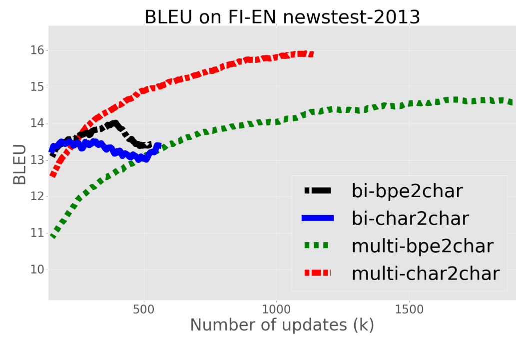

Figure 2: Multilingual models overfit less than bilingual models on low-resource language pairs.

Our empirical results show that the fully character-level model performs as well as, or better than, subword-level translation models. The performance gap is distinctly pronounced in the multilingual many-to-one translation task, where results show that character-level model can assign model capacities to different languages more efficiently than the subword-level models. We observe a particularly large improvement in FI-EN translation when the model was trained to translate multiple languages, indicating positive cross-lingual transfer to a low-resource language pair.

We discover two main benefits of the multilingual character-level model: (1) it is much more parameter efficient than the bilingual models and (2) it can naturally handle intra-sentence code-switching as a result of the many-to-one translation task.

This work presents a case for fully character-level translation: that translation at the level of character is strongly beneficial and should be encouraged more.

In the next stage of this research, we will investigate extending our multilingual many-to-one translation models to perform many-to-many translation, which will allow the decoder, similarly with the encoder, to learn from multiple target languages. Furthermore, a more thorough investigation into model architectures and hyperparameters is needed.

Acknowledgements

KC thanks the support by Facebook, Google (Google Faculty Award 2016) and NVidia (GPU Center of Excellence 2015-2016). This work was partly supported by Samsung Advanced Institute of Technology (Neural Machine Translation). JL was supported by Qualcomm Innovation Fellowship, and thanks David Yenicelik and Kevin Wallimann for their contribution in designing the qualitative analysis. The authors would also like to thank Prof. Zheng Zhang (NYU Shanghai) for fruitful discussion and comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics, and Structure in Statistical Translation*, page 103.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the Empiricial Methods in Natural Language Processing*.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Marta R. Costa-Jussá and Josè A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 357.

Ferdinand de Saussure. 1916. *Course in General Linguistics*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multilingual neural machine translation. *arXiv preprint arXiv:1606.04164*.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2015. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ray S. Jackendoff. 1992. *Semantic Structures*, volume 18. MIT press.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in Neural Information Processing Systems (NIPS 2015)*, volume 28.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2015. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS 2015)*, volume 28.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 1357.

Yijun Xiao and Kyunghyun Cho. 2016. Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers. *arXiv preprint arXiv:1602.00367*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NIPS 2015)*, volume 28.

A Supplementary Examples

We show additional sample translations in five scenarios: spelling mistakes (Table 8), rare and long words (Table 9), nonce words (Table 10), morphological inflections (Table 11) and intra-sentence code-switching (Table 12).

(a) Spelling mistakes

DE ori	Die Premierminister Indiens und Japans trafen sich in Tokio .
DE src	Die Premireminster Indiens und Japans trafen sch in Tokio .
EN ref	India and Japan prime ministers meet in Tokyo .
bpe2char	The Premier Minster of India and Japan met in Tokyo .
char2char	The prime ministers of India and Japan met in Tokyo .

(b) Spelling mistakes

DE ori	Wir haben nichts zu verlieren .
DE src	Wir haben nichts zu verlirren .
EN ref	We have nothing to lose .
bpe2char	We have nothing to limit .
char2char	We have nothing to lose .

Table 8: Sample translations of German sentences with spelling mistakes.

(c) Rare words

DE src	Die Kluser-Ampel sichere sowohl Radfahrer als auch Busfahrgäste und die Bergle-Bewohner .
EN ref	The Kluser lights protect cyclists , as well as those travelling by bus and the residents of Bergle .
bpe2char	The cluster traffic lights secure both cyclists and bus passengers and mountain residents .
char2char	The Kluser traffic lights secure both cyclists and bus passengers and the mountain mountain residents .

(d) Rare words

DE src	In Swasiland leben 245000 Aidswaisen .
EN ref	In Swaziland , there are 245,000 AIDS orphans .
bpe2char	In Swaziland , 245,000 Aidswaisen live .
char2char	24500 AIDS orphans live in Swaziland .

(e) Long words

DE src	Bezirksschornsteinfegermeister .
EN ref	District chimney sweep master .
bpe2char	District of the district of the district .
char2char	Residential Chimney Mayors .

(f) Long words

DE src	Ich bin keine Einhundert-Dollar-Note , die jedem gefällt .
EN ref	I am not a hundred dollar bill to please all .
bpe2char	I am not a one-hundred dollar touch that everyone likes .
char2char	I am not a hundred dollar notes that everyone likes .

Table 9: Sample translations of German sentences with rare and long words.

(g) Nonce words

DE src	Vieles dreht sich bei Hirst um lebensverlängernde oder sterbenbeschleunigende Mittelchen .
EN ref	Much is about Hirst's life-prolonging or death-accelerating agents .
bpe2char	There are many things to do with brain around extending life-extra or accelerating medium-sized businesses .
char2char	Much revolves around life-extension or dying acceleration in Hirst .

Table 10: Sample translations of a German sentence with a nonce word.

(h) Morphology

DE src	Kokainabhängiger Anwalt , der Drogenboss vor polizeilicher Ermittlung warnte , muss ins Gefängnis .
EN ref	Cocaine-addict lawyer who tipped off Mr Big about police investigation must be jailed .
bpe2char	Cocaine lawyer , warning drug boss from police investigation , must in prison .
char2char	Lawyers who warned the drugs boss before police investigation must be prisoners .

(i) Morphology

DE src	Der US-Senat genehmigte letztes Jahr ein 90 Millionen Dollar teures Pilotprojekt , das 10.000 Autos umfasst hätte .
EN ref	The U.S. Senate approved a $ 90-million pilot project last year that would have involved about 10,000 cars .
bpe2char	Last year , the US Senate approved a $ 90 million pilot project , which would have covened 10,000 cars .
char2char	The US Senate approved a $ 90 million pilot project approved last year , which would have included 10,000 cars .

Table 11: Sample translations of German sentences with morphological inflections.

(j) Multilingual

Multi src	Es wird geschätzt , že kdyby všichni zdravě jedli a dostatečně se hýbali , 30% случаев рако можно было бы предотвратить .
EN ref	It is estimated that if everyone ate healthily and sufficiently moved , 30 % of cancer cases could be prevented .
bpe2char	It is estimated that if everyone eats and moves enough , 30 % of cases could be prevented .
char2char	It is estimated that if everyone had eaten and moved sufficiently , 30 % of cases could be prevented .

(k) Multilingual

Multi src	А коалиция I-95 , která zahrnuje 17 státních ministerstev dopravy штатов восточного побережья (einschließlich Мэриленд , Пенсильванию , Вирджинию und Флориду) , изучает , как они смогут провести эти изменения .
EN ref	And the I-95 Coalition , which includes 17 state transportation departments along the Eastern Seaboard (including Maryland , Pennsylvania , Virginia and Florida) , is studying how they could go about implementing the change .
bpe2char	And the I-95 coalition , which includes 17 State ministries of the eastern coast (including Merilend , Pennsylvania , Virginia and Florida) , studies how they will be able to conduct these changes .
char2char	A coalition I-95 , which includes 17 state ministries of transport of the transport states of the East Coast (including Maryland , Pensinence , Virginia and Florida) , studies how they can do these changes .

(l) Multilingual

Multi src	Bei der , Metropolitního výboru pro dopravu für das Gebiet der San Francisco Bay erklärten Beamte , der Kongress könne das Problem des bankrotten Highway Trust Fund einfach, , подняв налоги на бензин .
EN ref	At the Metropolitan Transportation Commission in the San Francisco Bay Area , officials say Congress could very simply deal with the bankrupt Highway Trust Fund by raising gas taxes .
bpe2char	At the Metropolitan Committee on Transport for San Francisco Bay , officials declared that Congress could raise the problem of the bankrupt Highway Trust Fund easier .
char2char	At the Metropolitan Committee on Transport for the territory of San Francisco Bay , officials explained that Congress can simply raise the problem of the bankrupt Highway Trust Fund .

Table 12: Sample translations of mixed sentences from German (green), Czech (yellow) and Russian (blue) into English.