



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Understanding Pure Character-Based Neural Machine Translation: The Case of Translating Finnish into English

Citation for published version:

Tang, G, Sennrich, R & Nivre, J 2020, Understanding Pure Character-Based Neural Machine Translation: The Case of Translating Finnish into English. in *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 4251-4262, The 28th International Conference on Computational Linguistics, Virtual conference, 8/12/20. <<https://www.aclweb.org/anthology/2020.coling-main.375>>

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Publisher's PDF, also known as Version of record

Published In:

Proceedings of the 28th International Conference on Computational Linguistics

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Understanding Pure Character-Based Neural Machine Translation: The Case of Translating Finnish into English

Gongbo Tang¹ Rico Sennrich^{2,3} Joakim Nivre¹

¹Department of Linguistics and Philology, Uppsala University

²Institute of Computational Linguistics, University of Zurich

³School of Informatics, University of Edinburgh

firstname.lastname@{lingfil.uu.se, ed.ac.uk}

Abstract

Recent work has shown that deeper character-based neural machine translation (NMT) models can outperform subword-based models. However, it is still unclear what makes deeper character-based models successful. In this paper, we conduct an investigation into pure character-based models in the case of translating Finnish into English, including exploring the ability to learn word senses and morphological inflections and the attention mechanism. We demonstrate that word-level information is distributed over the entire character sequence rather than over a single character, and characters at different positions play different roles in learning linguistic knowledge. In addition, character-based models need more layers to encode word senses which explains why only deeper models outperform subword-based models. The attention distribution pattern shows that separators attract a lot of attention and we explore a sparse word-level attention to enforce character hidden states to capture the full word-level information. Experimental results show that the word-level attention with a single head results in 1.2 BLEU points drop.

1 Introduction

Neural machine translation (NMT) has boosted machine translation significantly in recent years (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). However, it is still unclear how NMT models work due to the black-box nature of neural networks. Better understandings of NMT models could guide us in improving NMT systems. Currently most of the studies towards understanding NMT models only take into account subword-based (e.g. BPE-based) models. Deeper character-based (CHAR) models have been shown to perform better than BPE-based models (Cherry et al., 2018). In this paper, we try to investigate the working mechanism of CHAR models. We explore the ability of CHAR models to learn word senses and morphological inflections and the attention mechanism.

Previous studies have tried to interpret and understand NMT models by interpreting attention weights (Ghader and Monz, 2017; Raganato and Tiedemann, 2018; Tang et al., 2018; Tang et al., 2019), using gradients (He et al., 2019), applying layer-wise relevance propagation (Ding et al., 2017), probing classification tasks (Belinkov et al., 2017b; Belinkov et al., 2017a; Belinkov et al., 2020; Poliak et al., 2018; Tang et al., 2019), and more intrinsic analysis (Ghader and Monz, 2019; Voita et al., 2019). However, only Belinkov et al. (2017a; Belinkov et al. (2020) have probed character-based representations. Belinkov et al. (2017a) have only explored character-aware word-level representations, while we investigate fully character-level representations, which are also studied in Belinkov et al. (2020). We apply more composition methods to explore how CHAR models learn linguistic knowledge and how attention extracts features directly from characters.

Probing classification tasks (Belinkov et al., 2017a) have emerged as a popular method to interpret the internal representations from neural networks. Given a probing classifier, the input is usually the representation of a word and the output is the corresponding linguistic tag. CHAR models pose new challenges for interpretability, and we investigate whether we can probe CHAR models in a way similar

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

to (sub)word-based models. In addition, can we extract word sense and morphological information about the full word from individual hidden states, or is this information distributed across multiple states? This has implications for interpreting neural CHAR models, but can also inform novel architectures, such as sparse attention mechanisms. Thus we first investigate the ability of CHAR models to learn word senses and morphology in Section 3. We apply different methods to compose information from characters and demonstrate that the word-level information is distributed over all the characters but characters at different positions play different roles in learning linguistic knowledge. We also explore the effect of encoder depth to answer why CHAR models outperform BPE-based models only when they have the settings with deeper encoder. The probing results show that CHAR models need more layers to learn word senses. Then in Section 4, we move on to explore the attention mechanism. The distribution pattern shows that separators attract much more attention compared to other characters. To study the effect of enforcing characters to capture the full word-level information, we investigate a sparse attention mechanism, i.e. a model that only attends to separators, which can be viewed as a word-level attention. The BLEU score drops 1.2 points when we apply the word-level sparse attention. This implies that only attending to separators by a single attention head is workable but not enough to extract all the necessary information.

The main findings are summarized as follows:

- Word sense and morphological information is distributed over all the characters, but characters at different positions play different roles in learning linguistic knowledge.
- CHAR models need more layers to encode word senses, which explains why only deeper models outperform BPE-based models.
- Separators attract much more attention compared to other characters; we find that only attending to separators with a single head attention is workable but not sufficient for translation.

2 Experiments

As RNN-based/CHAR models can in principle achieve state-of-the-art performance in NMT (Chen et al., 2018; Cherry et al., 2018) and most of analysis of NMT models are based on BPE-based models, we are interested in analyzing the working mechanisms of pure RNN-based CHAR models.¹ We follow Cherry et al. (2018) in using RNN-based models, and we focus on Finnish→English (FI→EN), because training CHAR models requires huge computational resources.

We first train CHAR models with different encoder depths and a BPE-based model for comparison. Then we explore how CHAR models learn word senses and morphological inflections via probing classification tasks, using representations generated by the trained models. For the morphological probing tasks, the classifiers predict the morphological tag given the representation of a token. For the word sense disambiguation (WSD) probing task where learning word senses is needed, the input to the classifier and the output are different from the classifiers in the morphological probing tasks. Instead, the representations of an ambiguous word and its candidate translation are both fed into the classifier and then the classifier predicts whether the candidate translation is correct or not.

2.1 Data

We train NMT models on the WMT15 shared task data (Bojar et al., 2015) for FI→EN to be able to compare with Cherry et al. (2018). There are about 2.1M sentence pairs in the training set after preprocessing with Moses scripts.

For the WSD probing task, we use the FI→EN part of the *MuCoW* (Raganato et al., 2019) test set, which is a multilingual test suite for WSD in the WMT19 shared task. It has 2,117 annotated sentences. Each annotation provides the ambiguous Finnish word, the domain of the sentence, and a set of translation candidates of the ambiguous word including both correct and incorrect translations. For each ambiguous word from an annotation, we generate multiple instances that are labeled with one translation candidate and a binary value indicating whether it corresponds to the correct sense. 1,000/1,000 instances are randomly selected as the development/test sets, and the remaining 6,325 instances are used for training

¹Also, training Transformer models at the character level with a large batch size requires enormous amounts of memory.

Feature	POS	Grammatical case	Locative case	Number	Infinitive	Voice
# Tag	16	4	8	8	3	2
Training	11,678	5,147	1,495	7,603	301	1,723
Dev/Test	1,500	650	200	1,000	40	200

Table 1: Statistics of in-domain data from *MuCoW* for morphological probings.

the probing classifiers.²

To extend *MuCoW* for the morphological probing tasks, we use the RNNTagger,³ which is trained on FinnTreebank²⁴ to generate the morphological tags. Finnish is a morphologically rich language, and in addition to *POS*, we generate data for 5 other morphological features: *grammatical case*, *locative case*, *number*, *infinitive*, and *voice*. These features vary in the types of tag. The data is roughly split into training/development/test sets at the ratio of 8:1:1. Each data entry for the probing tasks contains the representation of a token and the morphological tag. The detailed statistics are provided in Table 1.

2.2 Experimental Settings

NMT models We use the *Sockeye* (Hieber et al., 2017) toolkit to train NMT models. The encoder is a stack of 1 bidirectional RNN and 6 unidirectional RNNs, and the decoder has 8 unidirectional RNNs. We choose long short-term memory (LSTM) RNN unit (Hochreiter and Schmidhuber, 1997). The size of embeddings and hidden units is 512. We tie the source, target, and output embeddings. The beam size is 8 during inference. We employ the models that have the best perplexity on the validation set for evaluation. BLEU scores (Papineni et al., 2002) are computed by *sacrebleu* (Post, 2018). For CHAR models, we add separators between any two tokens including punctuation marks, and input character sequences to the model directly. The character vocabulary size is 379. For the BPE-based model, we learn a joint BPE model with 32K subwords (Sennrich et al., 2016). As Cherry et al. (2018) have shown that the depth is crucial to the success of CHAR models, we train a 4-layer CHAR model to study the effect of depth.

Probing classifiers These probing classifiers are feed-forward neural networks with only one hidden layer, using ReLU non-linear activation. The size of the hidden layer is set to 512. We use the Adam learning algorithm (Kingma and Ba, 2015). The classifiers are trained using a cross-entropy loss. Each classifier is trained for 180/100 epochs in the WSD/morphological probing tasks and the one that performs best on the development set is selected for evaluation. We train 5 times with different seeds for each classifier and report average accuracy.

In contrast to word-level hidden states, a word consists of multiple character-level hidden states in CHAR models. We are interested in how the word-level information, including word senses and morphological inflections, is distributed over the character hidden states, in a single state or spread over all hidden states. Thus we explore the following methods for composition:

- mean pooling: mean of hidden states
- max pooling: max of hidden states in each dimension
- last pooling: last hidden state
- first pooling: first hidden state
- randLSTM: output of a randomly initialized LSTM, whose input are the hidden states of characters; we use a 1-layer bidirectional LSTM with parameters initialized uniformly at random from $[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]$, where d is the hidden size of the LSTM (Wieting and Kiela, 2019).⁵

The *mean* pooling method, which simply averages all the hidden states of a word, can tell us how much word sense or morphological information has been encoded into the word and serves as the baseline. The

²Note that *MuCoW* has both in-domain (news and books) and out-of-domain data (subtitles), and the results are only based on the in-domain data unless otherwise specified. The ratio of in-domain data and out-of-domain data is around 1:2.

³<https://www.cis.uni-muenchen.de/~schmid/tools/RNNTagger/>

⁴<http://www.ling.helsinki.fi/kieliteknologia/tutkimus/treebank/index.shtml>

⁵Note that we first use randLSTM to generate the composition, then we feed the output into the following classifiers. That is to say, the parameters of the randLSTM are fixed during the training of the following probing classifiers. This can reduce the effect of LSTMs on the probing classifiers.

System	Cherry et al. (2018)	bpe-d7	char-d4	char-d7
Encoder Depth	6	7	4	7
BLEU	19.5	16.9	16.3	17.2

Table 2: BLEU scores of the NMT models on FI→EN.

Feature	model layer	char-d7			bpe-d7			Feature	model layer	char-d7			bpe-d7		
		1	3	5	1	3	5			1	3	5	1	3	5
WSD	mean	71.95	79.94	81.57	83.02	84.53	83.77	POS	mean	93.29	93.91	94.80	41.29	42.07	40.79
	max	71.76	74.91	75.60	84.65	85.28	83.52		max	92.23	93.56	94.20	43.35	44.35	42.31
	last	74.78	80.25	80.88	82.58	84.59	84.40		last	73.81	81.37	87.43	40.08	39.83	39.43
	first	74.28	81.89	82.96	84.40	85.66	83.65		first	92.08	91.92	91.43	39.28	40.91	39.19
	randLSTM	74.65	83.90	82.89	82.26	80.44	81.07		randLSTM	92.57	92.83	92.56	35.91	35.52	35.92
Grammatical	mean	60.40	58.37	58.15	66.03	64.12	61.75	Locative	mean	26.00	25.60	27.30	47.90	45.40	40.30
	max	59.97	57.94	58.28	67.14	63.23	60.03		max	26.20	26.80	24.60	49.30	46.60	41.20
	last	57.94	54.68	56.55	66.62	64.09	61.35		last	24.40	21.30	24.50	50.70	47.10	40.10
	first	55.78	54.06	55.54	65.66	63.63	59.29		first	24.00	23.40	20.80	48.40	42.00	34.40
	randLSTM	92.25	90.62	87.51	55.88	56.34	53.66		randLSTM	81.60	75.10	78.30	30.70	32.00	27.80
Number	mean	75.10	75.26	75.34	72.38	73.54	70.98	Infinitive	mean	53.50	57.00	58.50	55.50	53.50	51.50
	max	73.92	73.90	73.50	72.82	73.02	71.10		max	58.50	52.50	54.00	59.00	57.50	61.50
	last	71.90	72.04	72.10	72.54	72.40	70.56		last	56.50	57.00	53.00	58.00	55.50	51.50
	first	72.94	73.06	72.90	72.60	72.38	69.88		first	50.00	47.50	49.00	57.50	52.50	54.00
	randLSTM	95.32	95.52	94.12	70.10	69.68	70.00		randLSTM	80.00	75.00	75.00	55.00	55.00	55.00
Voice	mean	84.00	83.70	84.30	85.60	85.70	84.20		mean	84.00	84.00	84.00	85.80	86.00	85.20
	max	84.00	84.00	84.00	85.80	86.00	85.20		max	82.40	84.10	83.10	83.80	83.60	83.00
	last	82.40	84.10	83.10	83.80	83.60	83.00		last	81.10	83.60	83.10	84.60	84.80	83.40
	first	81.10	83.60	83.10	84.60	84.80	83.40		first	94.50	95.00	95.20	84.00	82.90	83.00
	randLSTM	94.50	95.00	95.20	84.00	82.90	83.00		randLSTM						

Table 3: Accuracy (%) on WSD and morphological probing tasks using hidden states from *char-d7* and *bpe-d7*, with different composition methods. The numbers in bold are the best accuracy of each model in each probing task. The results of *char-d7* on morphological probing tasks consider separators as the last character of a word while those on the WSD probing tasks do not take separators into account.

first/last pooling method detects how much word sense or morphological information can be captured by a single character. *randLSTM* can test whether word sense or morphological information need to be modeled by a more complicated composition method or has been encoded into each hidden state and only need a simple *mean* pooling composition. Note that we are not pursuing better composition methods for probing tasks but investigating how CHAR models encode the word sense and morphological information into hidden states of characters. We also apply these composition methods to subwords.

2.3 Results

2.3.1 BLEU Scores of NMT Models

Table 2 gives the BLEU scores of the three NMT models, which are used for the following investigation. In accordance with Cherry et al. (2018), our deeper CHAR model (*char-d7*) indeed outperforms the BPE-based model with the same number of layers (*bpe-d7*). The CHAR model with 4 layers (*char-d4*) is inferior to the other two models as expected. The result of Cherry et al. (2018) in Table 2 is obtained using 6 bidirectional gated recurrent units (GRUs) (Cho et al., 2014) in the encoder, which are 12 unidirectional LSTMs. Our encoder only has 1 bidirectional LSTM and 6 unidirectional LSTMs. In addition, we do not apply label smoothing technique in our models. We assume that these differences in settings cause the performance gap. Nevertheless, we focus on exploring how CHAR models work rather than pursuing better performance.

2.3.2 Accuracy in Probing Tasks

Table 3 gives the accuracy in the WSD probing task and different morphological probing tasks, using hidden states from *char-d7* and *bpe-d7* with different composition methods. Results of *bpe-d7* are given for comparison. The bold numbers are the best accuracy of each model in each probing task.

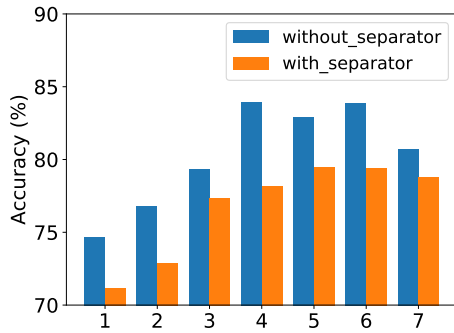


Figure 1: Accuracy of each layer from *char-d7* on the WSD probing task, considering separators or not, using *randLSTM* for composition.

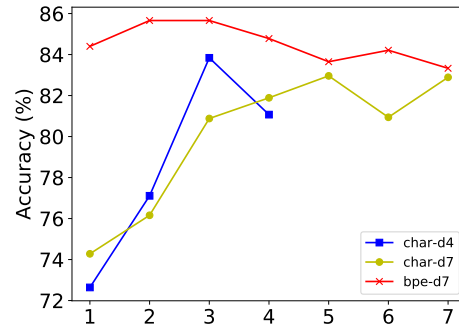


Figure 2: Accuracy evolution over layers on the WSD probing task, using *first* pooling for composition.

For the WSD probing task, we can see that using the first hidden state of the ambiguous words from the 3rd layer of *bpe-d7* achieves the highest accuracy (85.66%). In *char-d7*, hidden states from higher layers tends to perform better than those from lower layers when using pooling methods for composition but not when using *randLSTM* for composition.

We can tell that *char-d7* achieves significantly better performance than *bpe-d7* on all the morphological probing tasks, which is consistent with previous finding that character-level hidden states are better for learning morphology (Lee et al., 2017; Belinkov et al., 2017a; Durrani et al., 2019; Belinkov et al., 2020). In the *POS* probing task, the hidden states of characters are much superior to those of subwords.⁶ This indicates that the *POS* information is better encoded into hidden states than the other morphological features. In the *locative* probing task, although *char-d7* using *randLSTM* performs much better than *bpe-d7*, the performance of pooling methods is far behind that of *bpe-d7*. We interpret this as showing that *locative* case information is not simply distributed over character hidden states and we need a more complicated composition method to extract the information. Both *char-d7* and *bpe-d7* perform well on the *voice* probing task which is a relatively easy binary classification task.

3 Learning Linguistics

In this section, we interpret the ability of CHAR models to encode word senses and morphological inflections, by exploring the effect of separators, composition methods, the evolution over layers, and the robustness to domain-mismatch. We only analyze the encoder hidden states in the context of probing tasks. As the attention extracts features from encoder hidden states in a different way from our composition methods – pooling and *randLSTM*, the findings may not apply to the entire CHAR model.

3.1 Learning Word Senses

3.1.1 The Effect of Separators

Separators indicate word boundaries in the input sequences, which potentially are viewed as the end of a word by the CHAR model. We test the role of separators in learning word senses by testing the effect on the WSD probing task. Figure 1 displays the WSD accuracy of representations from different encoder layers with and without considering separators as the last character of a word, using *randLSTM* for composition. We can see that considering separators results in a lower accuracy which means that separators have a negative effect on learning word senses when we compose all the characters. However, the information of characters from a word passes to separators in the encoder. Separators do carry some of the word sense information and they can achieve up to 79.8% in accuracy. We speculate that separators

⁶We have noted the exceptionally low accuracy of *bpe-d7*, but we have not found any errors from the perspective of experimental settings.

Model	<i>char-d7</i>	<i>bpe-d7</i>
in-domain	83.9%	85.7%
out-of-domain	45.7%	49.1%

Table 4: Best accuracy of *char-d7* and *bpe-d7* on the WSD probing task, on in-domain and out-of-domain test sets.

also capture some morphological information which confuses the classifier in identifying word senses, and we will explore it in Section 3.2. Note that the following results in this section do not consider separators.

3.1.2 Composition Methods

Even though *first* pooling only utilizes the first hidden state of a word, it performs better than *mean* and *max* which use all the hidden states. We can infer that this CHAR model encodes more word sense information into the first characters of words. However, *randLSTM* can achieve higher accuracy than *first*. We conclude that this CHAR model also distributes the word sense information to other characters but we need a more complicated composition method to extract more word sense information.

For *bpe-d7*, *first* achieves the best accuracy among all the composition methods. We can tell that both the CHAR and BPE-based models encode much sense information into the first character/subword but the first subword is enough to represent the word sense. Moreover, *randLSTM* performs worse than the simple pooling methods in *bpe-d7*, which indicates that the information about word senses has been well represented by hidden states and we do not need a more complicated method to further extract it.

3.1.3 Evolution over Layers

Figure 2 shows the accuracy evolution over layers of both CHAR and BPE-based models, using *first* for composition. In the first layer, *char-d7* performs much worse than *bpe-d7*. However, the learning curve of CHAR models is much steeper than *bpe-d7*, especially in the first three layers. In the 7th layer, *char-d7* performs almost as well as *bpe-d7*. We speculate that it takes several layers for the first character to learn the basic sense of a word and it need more layers to learn the contextualized/disambiguated word sense. This can explain why previous shallow CHAR models do not perform well, as word senses are learned layer by layer.

3.1.4 Robustness to Domain-Mismatch

CHAR models have been shown robust to spelling mistakes, rare words, morphology, and compounds (Lee et al., 2017; Cherry et al., 2018). The meaning of an ambiguous word is likely to vary with domains. Thus, here we investigate the robustness to domain-mismatch of CHAR models when learning word senses. We directly test the models trained on in-domain data using the out-of-domain test set.

Table 4 gives the best accuracy on in-domain and out-of-domain test sets. The accuracy of both models has a substantial drop on the out-of-domain test set which is consistent with the finding from Raganato et al. (2019). The drop of *char-d7* is even bigger than that of *bpe-d7* which indicates that CHAR models are not more robust to domain-mismatch when learning word senses compared to BPE-based models.

3.2 Learning Morphology

3.2.1 The Effect of Separators

We have demonstrated that separators have a negative effect on the WSD probing task in Section 3.1. As separators indicate the word boundary information and morphological features are reflected over the middle parts or the last parts of a word in Finnish, here we hypothesize that these separators are important to the morphological probing tasks. We measure the effect of separators by comparing the performance on the morphological probing tasks whether considering separators as the last character of a word.

We find that the representations considering the separators are evidently superior. Figure 3 displays the comparison on the *grammatical* and *number* probing tasks and we get the same pattern in the other morphological probing tasks. These results indicate that separators capture much of the word-level

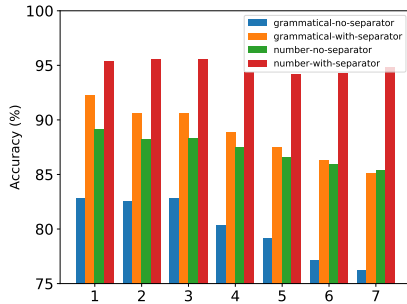


Figure 3: Accuracy on *grammatical* and *number* probing tasks, considering the separators or not, using *randLSTM* for composition.

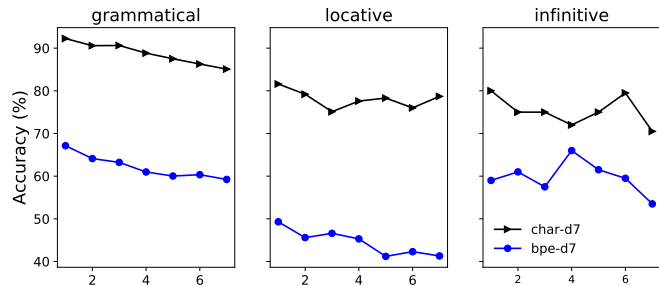


Figure 4: Accuracy evolution over layers on *grammatical*, *locative*, and *infinitive* probing tasks, in *char-d7* and *bpe-d7*, using *randLSTM* and *max* for composition, respectively.

morphological information which is not encoded into other characters. The following results in this section take separators into account.

3.2.2 Composition Methods

In Table 3, *max* also considers all the hidden states but has different performance compared to *mean*, especially on *infinitive* and *number*. *first* and *last* are inferior to *mean* and *max* which implies that the first and the last character of the word do not capture all the word-level morphological information, even though we have shown that the last character (separator) has some crucial information. In particular, *last* only achieves 88.16% on the POS probing task while *mean* achieves 94.80%. Thus, we conclude that the model has not learned to build a full morphological representation of words at the first or last position, but that the information remains distributed across positions. In *bpe-d7*, *max* pooling achieves the best results in 4 out of 6 probing tasks. *first* and *last* are usually inferior to *mean/max*.

randLSTM performs significantly better than *mean/max* except on the *POS* probing task, even though the LSTM is randomly initialized without any training. The gaps vary from 10.9% to 54.1%, especially in the probing task on *locative*. We can infer that the information of word structure is not well encoded into hidden states, thus we need a more complicated composition method to abstract the information.

For *POS*, *randLSTM* performs worse than *mean/max*. We attribute this to the fact that *POS* is a global feature compared to other morphological features and has been well encoded into hidden states. Thus, it does not need further extraction. In contrast to the results for *char-d7*, *randLSTM* for *bpe-d7* is not as good as the simple pooling methods in any of the probing tasks. We infer that the morphological information has been well encoded into subword hidden states and does not need further composition.

3.2.3 Evolution over Layers

Figure 4 exhibits the evolution of learning *grammatical*, *locative*, and *infinitive* features over layers. We can see that the overall accuracy of *char-d7* and *bpe-d7* tends to go down over layers (also in other morphological probing tasks). The results are consistent with the previous findings in Belinkov et al. (2017a; Belinkov et al. (2017b; Belinkov et al. (2020) that hidden states from the lower layers are better at learning morphology.

3.2.4 Effect of Encoder Depth

char-d7 is superior to *char-d4* in the translation task but it is still not clear how hidden states from NMT models with different encoder depths perform on the morphological probing tasks. As the hidden states in the last layer are fed to the decoder in the character-level NMT models, we first compare the hidden states from the last layer of both models, i.e. the 7th layer of *char-d7* and the 4th layer of *char-d4*. Figure 5 displays the performance in all the morphological probing tasks. Generally, *char-d7* outperforms *char-d4* on the probing task, with the exception of the *infinitive* probing task. However, note that a comparison of the last encoder layer does not tell the full story. Looking at the evolution of probing performance over

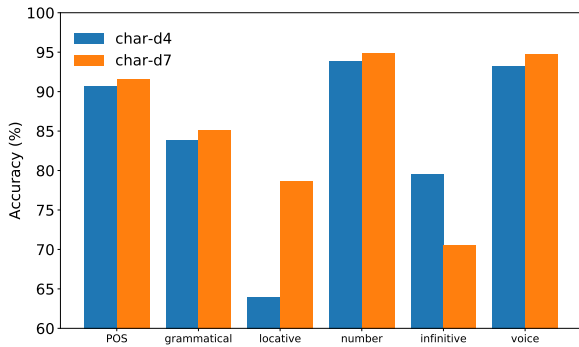


Figure 5: Accuracy on the six morphological probing tasks, using the last layer from *char-d4* and *char-d7*.

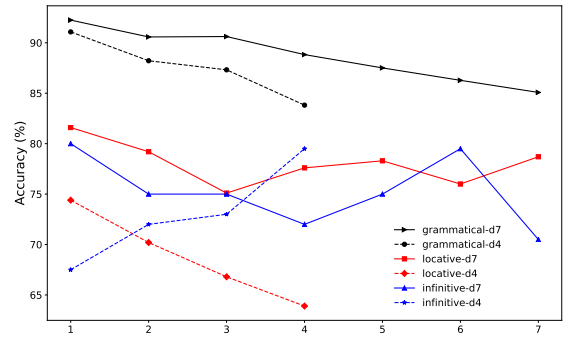


Figure 6: Accuracy evolution over layers of *char-d4* and *char-d7*, on *grammatical*, *locative*, and *infinitive* probing tasks.

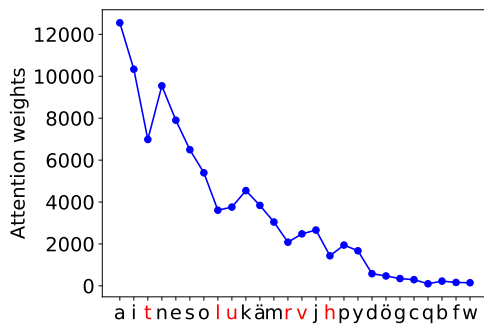


Figure 7: Attention distributions over characters in *char-d7* trained on FI→EN, summing up all the attention weights over each character. The characters are sorted in descending order of frequency and the characters in red are some exceptions in the trend curve.

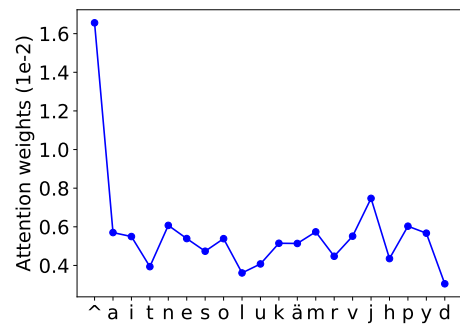


Figure 8: Attention weight over the 20 most frequent characters in *char-d7* trained on FI→EN, averaging all the attention weights over each character. “^” denotes the separator.

layers (Figure 6), we can see that *char-d7* typically achieves the highest probing accuracy at the first layer, outperforming *char-d4*.

4 Attention Mechanisms

In the encoder-decoder attention (Bahdanau et al., 2015; Luong et al., 2015), a higher attention weight means that the source token contributes more to the prediction at current step. Thus, we could utilize the attention distributions to explore how CHAR models pay attention to the source characters during translation.

CHAR models encode information into a longer sequence which essentially increases the representational capacity of the encoder. We are interested in exploring the effect of restricting the capacity of the encoder states that are passed to the decoder. Thus, we apply the word-level attention which attends to a character of each word and potentially enforces the character captures the full word information.

4.1 Attention Distributions over Characters

We explore the attention distributions that are generated when translating *newstest2015*. In addition to all the characters, we also consider the separator character. We calculate the attention weights over each source character. It is interesting that the sum of attracted attention is basically consistent with the frequency of characters in the source language (Finnish) which is shown in Figure 7. This pattern indicates that most of the characters are treated equally during the overall decoding. However, when we average all the attention weights over each character and compare the separator with other characters, the

Model	BLEU	Drop
char-d7	16.0	1.2
bpe-d7	14.3	2.6

Table 5: Results of applying word-level attention to both CHAR and BPE-based models.

separator apparently attracts much more attention as shown in Figure 8, which illustrates the attention weights over the 20 most frequent characters.

We also count the frequency of attracting the most attention at each decoding step for all the source characters. We find that the separator accounts for 31.4% of all the characters that have the highest weights. In addition, there are 29.6% of normal characters and 39.6% of separators in the target-side that distribute the highest attention weight to a source separator, which indicates that a large portion of normal characters also extract most of features from separators.

The separators attract a lot of attention which is similar to the attention patterns of BERT (Devlin et al., 2019) found by Clark et al. (2019). However, they find that the attention to the separators is used as a “no-op” for attention heads. In our settings, there is only one layer attention with one attention head, and we cannot regard the attention to separators as a “no-op”. Since the separators make the word representations better in morphology, we argue that the separators between words in the character sequences are encoded with rich linguistic features and contribute to the translation, which is different from the separators between sentences in BERT.

4.2 Word-Level Attention

As we have shown that separators have captured some linguistic knowledge, we enforce the word-level attention only attends to separators. We retrain the models with word-level attention from scratch. We also apply the word-level attention to BPE-based models as comparison. In that case, the attention attends to the last subword of a word.

The BLEU scores of the models with word-level attention are given in Table 5. The BLEU scores drop in both *char-d7* and *bpe-d7*. It indicates that restricting the capacity of encoder states that are passed to the decoder has a negative effect on the BLEU scores. The smaller drop on *char-d7* means that the word-level information can be better extracted from the character-level hidden states compared to the subword-level hidden states. *char-d7* does not perform too badly despite only attending to the separators. This indicates that the attention mechanism is very flexible and could force the model to encode more information into separators. However, since there is only one attention head, some information from the source is inevitably lost. It would be interesting to explore a multi-layer attention with multiple heads, such as Transformer (Vaswani et al., 2017) attention, where the information could be extracted from the other attention heads as well. We leave this as future work.

5 Conclusion

CHAR models have been shown to perform better than BPE-based models in NMT yet they pose new challenges for interpretability. In this paper, we investigate CHAR models via the WSD and six morphological probing tasks to learn how CHAR models learn word senses and morphology, in the case of translating Finnish into English. We also explore the attention distribution pattern and a sparse word-level attention to learn the working mechanism of attention.

In the probing tasks, we find that separators also have captured some linguistic knowledge. We apply different composition methods to the characters of a word, and we demonstrate that the word sense and morphological information is distributed over all the characters rather than some specific characters. Moreover, characters at different positions play different roles in learning linguistic knowledge. CHAR models are better at learning morphology but we need a more complicated composition method, such as a randomly initialized LSTM, to extract all the encoded information. These results on probing tasks show that we can extract word sense information and morphological features from character-level hidden states

and that these features are encoded in different ways. In addition, we explore the effect of encoder depth and show that CHAR models require more layers to encode word senses, which explains why only deeper CHAR models outperform BPE-based models. The attention distribution shows that separators attract a lot of attention, and we show that the sparse word-level attention only attending to separators is workable but not enough for translation.

As we have shown that characters at different positions specialize in learning word senses and morphology, it will be interesting to explore sparse attention with multiple heads in the future which could learn to extract features from different aspects.

Acknowledgments

We thank all reviewers for their valuable and insightful comments. We acknowledge the computational resources provided by CSC in Helsinki and Sigma2 in Oslo through NeIC-NLPL (www.nlpl.eu). GT is mainly funded by the Chinese Scholarship Council (NO. 201607110016).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.
- Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the Linguistic Representational Power of Neural Machine Translation Models. *Computational Linguistics*, 46(1):1–52.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.
- Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86. Association for Computational Linguistics.
- Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting Character-Based Neural Machine Translation with Capacity and Compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT’s Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, USA. Association for Computational Linguistics.
- Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and Understanding Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. Association for Computational Linguistics.
- Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One Size Does Not Fit All: Comparing NMT Representations of Different Granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1504–1516, Minneapolis, USA. Association for Computational Linguistics.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1243–1252, Sydney, Australia. The Proceedings of Machine Learning Research.
- Hamidreza Ghader and Christof Monz. 2017. What does Attention in Neural Machine Translation Pay Attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Hamidreza Ghader and Christof Monz. 2019. An Intrinsic Nearest Neighbor Analysis of Neural Machine Translation Architectures. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 107–117, Dublin, Ireland. European Association for Machine Translation.
- Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards Understanding Neural Machine Translation with Word Importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 952–961, Hong Kong, China. Association for Computational Linguistics.
- Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, USA. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA. Association for Computational Linguistics.
- Adam Poliak, Yonatan Belinkov, James Glass, and Benjamin Van Durme. 2018. On the Evaluation of Semantic Phenomena in Neural Machine Translation Using Natural Language Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 513–523, New Orleans, USA. Association for Computational Linguistics.
- Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics.

- Alessandro Raganato and Jörg Tiedemann. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.
- Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW Test Suite at WMT 2019: Automatically Harvested Multilingual Contrastive Word Sense Disambiguation Test Sets for Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480, Florence, Italy. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.
- Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Brussels, Belgium. Association for Computational Linguistics.
- Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019. Encoders Help You Disambiguate Word Senses in Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1429–1435, Hong Kong, China. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.
- Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4387–4397, Hong Kong, China. Association for Computational Linguistics.
- John Wieting and Douwe Kiela. 2019. No Training Required: Exploring Random Encoders for Sentence Classification. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA.