



THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### From characters to words to in between: Do we capture morphology?

**Citation for published version:**

Vania, C & Lopez, A 2017, From characters to words to in between: Do we capture morphology? in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 2016-2027, 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30/07/17. <https://doi.org/10.18653/v1/P17-1184>

**Digital Object Identifier (DOI):**

[10.18653/v1/P17-1184](https://doi.org/10.18653/v1/P17-1184)

**Link:**

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

**Document Version:**

Peer reviewed version

**Published In:**

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)

**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](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# From Characters to Words to in Between: Do We Capture Morphology?

Clara Vania and Adam Lopez

Institute for Language, Cognition and Computation

School of Informatics

University of Edinburgh

c.vania@ed.ac.uk, alopez@inf.ed.ac.uk

## Abstract

Words can be represented by composing the representations of subword units such as word segments, characters, and/or character n-grams. While such representations are effective and may capture the morphological regularities of words, they have not been systematically compared, and it is not understood how they interact with different morphological typologies. On a language modeling task, we present experiments that systematically vary (1) the basic unit of representation, (2) the composition of these representations, and (3) the morphological typology of the language modeled. Our results extend previous findings that character representations are effective across typologies, and we find that a previously unstudied combination of character trigram representations composed with bi-LSTMs outperforms most others. But we also find room for improvement: none of the character-level models match the predictive accuracy of a model with access to true morphological analyses, even when learned from an order of magnitude more data.

## 1 Introduction

Continuous representations of words learned by neural networks are central to many NLP tasks (Cho et al., 2014; Chen and Manning, 2014; Dyer et al., 2015). However, directly mapping a finite set of word types to a continuous representation has well-known limitations. First, it makes a closed vocabulary assumption, enabling only generic out-of-vocabulary handling. Second, it cannot exploit systematic functional relationships in learning. For example, *cat* and *cats* stand in the

same relationship as *dog* and *dogs*. While this relationship might be discovered for these specific frequent words, it does not help us learn that the same relationship also holds for the much rarer words *sloth* and *sloths*.

These functional relationships reflect the fact that words are composed from smaller units of meaning, or morphemes. For instance, *cats* consists of two morphemes, *cat* and *-s*, with the latter shared by the words *dogs* and *tarsiers*. Modeling this effect is crucial for languages with rich morphology, where vocabulary sizes are larger, many more words are rare, and many more such functional relationships exist. Hence, some models produce word representations as a function of subword units obtained from morphological segmentation or analysis (Luong et al., 2013; Botha and Blunsom, 2014; Cotterell and Schütze, 2015). A downside of these models is that they depend on morphological segmenters or analyzers.

Morphemes typically have similar orthographic representations across words. For example, the morpheme *-s* is realized as *-es* in *finches*. Since this variation is limited, the general relationship between morphology and orthography can be exploited by composing the representations of characters (Ling et al., 2015; Kim et al., 2016), character n-grams (Sperr et al., 2013; Wieting et al., 2016; Bojanowski et al., 2016; Botha and Blunsom, 2014), bytes (Plank et al., 2016; Gillick et al., 2016), or combinations thereof (Santos and Zadrozny, 2014; Qiu et al., 2014). These models are compact, can represent rare and unknown words, and do not require morphological analyzers. They raise a provocative question: Does NLP benefit from models of morphology, or can they be replaced entirely by models of characters?

The relative merits of word, subword, and character-level models are not fully understood because each new model has been compared on

different tasks and datasets, and often compared against word-level models. A number of questions remain open:

1. How do representations based on morphemes compare with those based on characters?
2. What is the best way to compose subword representations?
3. Do character-level models capture morphology in terms of predictive utility?
4. How do different representations interact with languages of different morphological typologies?

The last question is raised by [Bender \(2013\)](#): languages are typologically diverse, and the behavior of a model on one language may not generalize to others. Character-level models implicitly assume concatenative morphology, but many widely-spoken languages feature non-concatenative morphology, and it is unclear how such models will behave on these languages.

To answer these questions, we performed a systematic comparison across different models for the simple and ubiquitous task of language modeling. We present experiments that vary (1) the type of subword unit; (2) the composition function; and (3) morphological typology. To understand the extent to which character-level models capture true morphological regularities, we present oracle experiments using human morphological annotations instead of automatic morphological segments. Our results show that:

1. For most languages, character-level representations outperform the standard word representations. Most interestingly, a previously unstudied combination of character trigrams composed with bi-LSTMs performs best on the majority of languages.
2. Bi-LSTMs and CNNs are more effective composition functions than addition.
3. Character-level models learn functional relationships between orthographically similar words, but don't (yet) match the predictive accuracy of models with access to true morphological analyses.
4. Character-level models are effective across a range of morphological typologies, but orthography influences their effectiveness.

word	tries
morphemes	try+s
morphs	tri+es
morph. analysis	try+VB+3rd+SG+Pres

Table 1: The morphemes, morphs, and morphological analysis of *tries*.

## 2 Morphological Typology

A **morpheme** is the smallest unit of meaning in a word. Some morphemes express core meaning (**roots**), while others express one or more dependent **features** of the core meaning, such as person, gender, or aspect. A **morphological analysis** identifies the lemma and features of a word. A **morph** is the surface realization of a morpheme ([Morley, 2000](#)), which may vary from word to word. These distinctions are shown in Table 1.

Morphological typology classifies languages based on the processes by which morphemes are composed to form words. While most languages will exhibit a variety of such processes, for any given language, some processes are much more frequent than others, and we will broadly identify our experimental languages with these processes.

When morphemes are combined sequentially, the morphology is **concatenative**. However, morphemes can also be composed by **non-concatenative** processes. We consider four broad categories of both concatenative and non-concatenative processes in our experiments.

**Fusional languages** realize multiple features in a single concatenated morpheme. For example, English verbs can express number, person, and tense in a single morpheme:

*wanted* (English)

*want + ed*

*want + VB+1st+SG+Past*

**Agglutinative languages** assign one feature per morpheme. Morphemes are concatenated to form a word and the morpheme boundaries are clear. For example ([Haspelmath, 2010](#)):

*okursam* (Turkish)

*oku+r+sa+m*

“read”+AOR+COND+1SG

**Root and Pattern Morphology** forms words by inserting consonants and vowels of dependent morphemes into a consonantal root based on a given pattern. For example, the Arabic root *ktb* (“write”) produces ([Roark and Sproat, 2007](#)):

*katab* “wrote” (Arabic)

*takaatab* “wrote to each other” (Arabic)

**Reduplication** is a process where a word form is produced by repeating part or all of the root to express new features. For example:

*anak* “child” (Indonesian)

*anak-anak* “children” (Indonesian)

*buah* “fruit” (Indonesian)

*buah-buahan* “various fruits” (Indonesian)

### 3 Representation Models

We compare ten different models, varying subword units and composition functions that have commonly been used in recent work, but evaluated on various different tasks (Table 2). Given word  $w$ , we compute its representation  $\mathbf{w}$  as:

$$\mathbf{w} = f(\mathbf{W}_s, \sigma(w)) \quad (1)$$

where  $\sigma$  is a deterministic function that returns a sequence of subword units;  $\mathbf{W}_s$  is a parameter matrix of representations for the vocabulary of subword units; and  $f$  is a composition function which takes  $\sigma(w)$  and  $\mathbf{W}_s$  as input and returns  $\mathbf{w}$ . All of the representations that we consider take this form, varying only in  $f$  and  $\sigma$ .

#### 3.1 Subword Units

We consider four variants of  $\sigma$  in Equation 1, each returning a different type of subword unit: character, character trigram, or one of two types of morph. Morphs are obtained from Morfessor (Smit et al., 2014) or a word segmentation based on Byte Pair Encoding (BPE; Gage (1994)), which has been shown to be effective for handling rare words in neural machine translation (Sennrich et al., 2016). BPE works by iteratively replacing frequent pairs of characters with a single unused character. For Morfessor, we use default parameters while for BPE we set the number of merge operations to 10,000.<sup>1</sup> When we segment into character trigrams, we consider all trigrams in the word, including those covering notional beginning and end of word characters, as in Sperr et al. (2013). Example output of  $\sigma$  is shown in Table 3.

#### 3.2 Composition Functions

We use three variants of  $f$  in Eq. 1. The first constructs the representation  $\mathbf{w}$  of word  $w$  by adding

<sup>1</sup>BPE takes a single parameter: the number of merge operations. We tried different parameter values (1k, 10k, 100k) and manually examined the resulting segmentation on the English dataset. Qualitatively, 10k gave the most plausible segmentation and we used this setting across all languages.

the representations of its subwords  $s_1, \dots, s_n = \sigma(w)$ , where the representation of  $s_i$  is vector  $\mathbf{s}_i$ .

$$\mathbf{w} = \sum_{i=1}^n \mathbf{s}_i \quad (2)$$

The only subword unit that we don’t compose by addition is characters, since this will produce the same representation for many different words.

Our second composition function is a bi-directional long-short-term memory (**bi-LSTM**), which we adapt based on its use in the character-level model of Ling et al. (2015) and its widespread use in NLP generally. Given  $\mathbf{s}_i$  and the previous LSTM hidden state  $\mathbf{h}_{i-1}$ , an LSTM (Hochreiter and Schmidhuber, 1997) computes the following outputs for the subword at position  $i$ :

$$\mathbf{h}_i = LSTM(\mathbf{s}_i, \mathbf{h}_{i-1}) \quad (3)$$

$$\hat{s}_{i+1} = g(\mathbf{V}^T \cdot \mathbf{h}_i) \quad (4)$$

where  $\hat{s}_{i+1}$  is the predicted target subword,  $g$  is the softmax function and  $\mathbf{V}$  is a weight matrix.

A bi-LSTM (Graves et al., 2005) combines the final state of an LSTM over the input sequence with one over the reversed input sequence. Given the hidden state produced from the final input of the forward LSTM,  $\mathbf{h}_n^{fw}$  and the hidden state produced from the final input of the backward LSTM  $\mathbf{h}_0^{bw}$ , we compute the word representation as:

$$\mathbf{w}_t = \mathbf{W}_f \cdot \mathbf{h}_n^{fw} + \mathbf{W}_b \cdot \mathbf{h}_0^{bw} + \mathbf{b} \quad (5)$$

where  $\mathbf{W}_f$ ,  $\mathbf{W}_b$ , and  $\mathbf{b}$  are parameter matrices and  $\mathbf{h}_n^{fw}$  and  $\mathbf{h}_0^{bw}$  are forward and backward LSTM states, respectively.

The third composition function is a convolutional neural network (**CNN**) with highway layers, as in Kim et al. (2016). Let  $c_1, \dots, c_k$  be the sequence of characters of word  $w$ . The character embedding matrix is  $\mathbf{C} \in \mathbb{R}^{d \times k}$ , where the  $i$ -th column corresponds to the embeddings of  $c_i$ . We first apply a narrow convolution between  $\mathbf{C}$  and a filter  $\mathbf{F} \in \mathbb{R}^{d \times n}$  of width  $n$  to obtain a feature map  $\mathbf{f} \in \mathbb{R}^{k-n+1}$ . In particular, the computation of the  $j$ -th element of  $\mathbf{f}$  is defined as

$$\mathbf{f}[j] = \tanh(\langle \mathbf{C}[:, j : j + n - 1], \mathbf{F} \rangle + b) \quad (6)$$

where  $\langle A, B \rangle = \text{Tr}(\mathbf{A}\mathbf{B}^T)$  is the Frobenius inner product and  $b$  is a bias. The CNN model applies filters of varying width, representing features

Models	Subword Unit(s)	Composition Function
Sperr et al. (2013)	words, character n-grams	addition
Luong et al. (2013)	morphs (Morfessor)	recursive NN
Botha and Blunsom (2014)	words, morphs (Morfessor)	addition
Qiu et al. (2014)	words, morphs (Morfessor)	addition
Santos and Zadrozny (2014)	words, characters	CNN
Cotterell and Schütze (2015)	words, morphological analyses	addition
Sennrich et al. (2016)	morphs (BPE)	none
Kim et al. (2016)	characters	CNN
Ling et al. (2015)	characters	bi-LSTM
Wieting et al. (2016)	character n-grams	addition
Bojanowski et al. (2016)	character n-grams	addition
Vylomova et al. (2016)	characters, morphs (Morfessor)	bi-LSTM, CNN
Miyamoto and Cho (2016)	words, characters	bi-LSTM
Rei et al. (2016)	words, characters	bi-LSTM
Lee et al. (2016)	characters	CNN
Kann and Schütze (2016)	characters, morphological analyses	none
Heigold et al. (2017)	words, characters	bi-LSTM, CNN

Table 2: Summary of previous work on representing words through compositions of subword units.

Unit	Output of $\sigma(wants)$
Morfessor	$\hat{w}ant, s\$$
BPE	$\hat{w}, ant\$$
char-trigram	$\hat{w}a, wan, ant, nts ts\$$
character	$\hat{, w, a, n, t, s, \$$
analysis	want+VB, +3rd, +SG, +Pres

Table 3: Input representations for *wants*.

of character n-grams. We then calculate the max-over-time of each feature map.

$$y_j = \max_j \mathbf{f}[j] \quad (7)$$

and concatenate them to derive the word representation  $\mathbf{w}_t = [y_1, \dots, y_m]$ , where  $m$  is the number of filters applied. Highway layers allow some dimensions of  $\mathbf{w}_t$  to be carried or transformed. Since it can learn character n-grams directly, we only use the CNN with character input.

### 3.3 Language Model

We use language models (LM) because they are simple and fundamental to many NLP applications. Given a sequence of text  $s = w_1, \dots, w_T$ , our LM computes the probability of  $s$  as:

$$P(w_1, \dots, w_T) = \prod_{t=1}^T P(y_t | w_1, \dots, w_{t-1}) \quad (8)$$

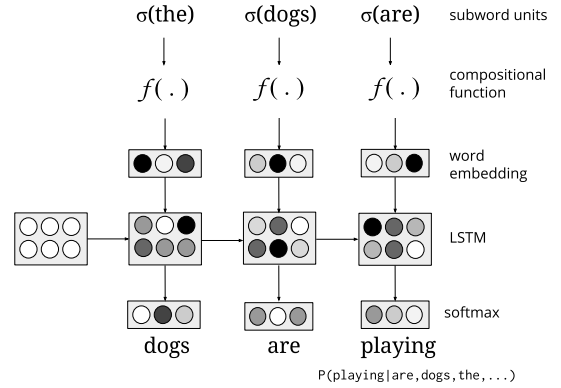


Figure 1: Our LSTM-LM architecture.

where  $y_t = w_t$  if  $w_t$  is in the output vocabulary and  $y_t = \text{UNK}$  otherwise.

Our language model is an LSTM variant of recurrent neural network language (RNN) LM (Mikolov et al., 2010). At time step  $t$ , it receives input  $w_t$  and predicts  $y_{t+1}$ . Using Eq. 1, it first computes representation  $\mathbf{w}_t$  of  $w_t$ . Given this representation and previous state  $\mathbf{h}_{t-1}$ , it produces a new state  $\mathbf{h}_t$  and predicts  $y_{t+1}$ :

$$\mathbf{h}_t = \text{LSTM}(\mathbf{w}_t, \mathbf{h}_{t-1}) \quad (9)$$

$$\hat{y}_{t+1} = g(\mathbf{V}^T \cdot \mathbf{h}_t) \quad (10)$$

where  $g$  is a softmax function over the vocabulary yielding the probability in Equation 8. Note that this design means that we can *predict* only words

Typology	Languages	#tokens	#types
Fusional	Czech	1.2M	125.4K
	English	1.2M	81.1K
	Russian	0.8M	103.5K
Agglutinative	Finnish	1.2M	188.4K
	Japanese	1.2M	59.2K
	Turkish	0.6M	126.2K
Root&Pattern	Arabic	1.4M	137.5K
	Hebrew	1.1M	104.9K
Reduplication	Indonesian	1.2M	76.5K
	Malaysian	1.2M	77.7K

Table 4: Statistics of our datasets.

from a finite output vocabulary, so our models differ only in their representation of context words. This design makes it possible to compare language models using perplexity, since they have the same event space, though open vocabulary word prediction is an interesting direction for future work.

The complete architecture of our system is shown in Figure 1, showing segmentation function  $\sigma$  and composition function  $f$  from Equation 1.

## 4 Experiments

We perform experiments on ten languages (Table 4). We use datasets from Ling et al. (2015) for English and Turkish. For Czech and Russian we use Universal Dependencies (UD) v1.3 (Nivre et al., 2015). For other languages, we use preprocessed Wikipedia data (Al-Rfou et al., 2013).<sup>2</sup> For each dataset, we use approximately 1.2M tokens to train, and approximately 150K tokens each for development and testing. Preprocessing involves lowercasing (except for character models) and removing hyperlinks.

To ensure that we compared models and not implementations, we reimplemented all models in a single framework using Tensorflow (Abadi et al., 2015).<sup>3</sup> We use a common setup for all experiments based on that of Ling et al. (2015), Kim et al. (2016), and Miyamoto and Cho (2016). In preliminary experiments, we confirmed that our models produced similar patterns of perplexities for the reimplemented word and character LSTM

models of Ling et al. (2015). Even following detailed discussion with Ling (p.c.), we were unable to reproduce their perplexities exactly—our English reimplementation gives lower perplexities; our Turkish higher—but we do reproduce their general result that character bi-LSTMs outperform word models. We suspect that different preprocessing and the stochastic learning explains differences in perplexities. Our final model with bi-LSTMs composition follows Miyamoto and Cho (2016) as it gives us the same perplexity results for our preliminary experiments on the Penn Treebank dataset (Marcus et al., 1993), preprocessed by Mikolov et al. (2010).

### 4.1 Training and Evaluation

Our LSTM-LM uses two hidden layers with 200 hidden units and representation vectors for words, characters, and morphs all have dimension 200. All parameters are initialized uniformly at random from -0.1 to 0.1, trained by stochastic gradient descent with mini-batch size of 32, time steps of 20, for 50 epochs. To avoid overfitting, we apply dropout with probability 0.5 on the input-to-hidden layer and all of the LSTM cells (including those in the bi-LSTM, if used). For all models which do not use bi-LSTM composition, we start with a learning rate of 1.0 and decrease it by half if the validation perplexity does not decrease by 0.1 after 3 epochs. For models with bi-LSTMs composition, we use a constant learning rate of 0.2 and stop training when validation perplexity does not improve after 3 epochs. For the character CNN model, we use the same settings as the *small model* of Kim et al. (2016).

To make our results comparable to Ling et al. (2015), for each language we limit the output vocabulary to the most frequent 5,000 training words plus an unknown word token. To learn to predict unknown words, we follow Ling et al. (2015): in training, words that occur only once are stochastically replaced with the unknown token with probability 0.5. To evaluate the models, we compute perplexity on the test data.

## 5 Results and Analysis

Table 5 presents our main results. In six of ten languages, character-trigram representations composed with bi-LSTMs achieve the lowest perplexities. As far as we know, this particular model has not been tested before, though it is similar

<sup>2</sup>The Arabic and Hebrew dataset are unvocalized. Japanese mixes Kanji, Katakana, Hiragana, and Latin characters (for foreign words). Hence, a Japanese character can correspond to a character, syllable, or word. The preprocessed dataset is already word-segmented.

<sup>3</sup>Our implementation of these models can be found at <https://github.com/claravania/subword-lstm-lm>

Language	word	character		char trigrams		BPE		Morfessor		%imp
		bi-lstm	CNN	add	bi-lstm	add	bi-lstm	add	bi-lstm	
Czech	41.46	34.25	36.60	42.73	<b>33.59</b>	49.96	33.74	47.74	36.87	18.98
English	46.40	43.53	44.67	45.41	<b>42.97</b>	47.51	43.30	49.72	49.72	7.39
Russian	34.93	28.44	29.47	35.15	<b>27.72</b>	40.10	28.52	39.60	31.31	20.64
Finnish	24.21	20.05	20.29	24.89	<b>18.62</b>	26.77	19.08	27.79	22.45	23.09
Japanese	98.14	98.14	<b>91.63</b>	101.99	101.09	126.53	96.80	111.97	99.23	6.63
Turkish	66.97	54.46	55.07	<b>50.07</b>	54.23	59.49	57.32	62.20	62.70	25.24
Arabic	48.20	42.02	43.17	50.85	<b>39.87</b>	50.85	42.79	52.88	45.46	17.28
Hebrew	38.23	31.63	33.19	39.67	<b>30.40</b>	44.15	32.91	44.94	34.28	20.48
Indonesian	46.07	45.47	46.60	58.51	45.96	59.17	<b>43.37</b>	59.33	44.86	5.86
Malay	54.67	53.01	<b>50.56</b>	68.51	50.74	68.99	51.21	68.20	52.50	7.52

Table 5: Language model perplexities on test. The best model for each language is highlighted in **bold** and the improvement of this model over the word-level model is shown in the final column.

to (but more general than) the model of Sperr et al. (2013). We can see that the performance of character, character trigrams, and BPE are very competitive. Composition by bi-LSTMs or CNN is more effective than addition, except for Turkish. We also observe that BPE always outperforms Morfessor, even for the agglutinative languages. We now turn to a more detailed analysis by morphological typology.

**Fusional languages.** For these languages, character trigrams composed with bi-LSTMs outperformed all other models, particularly for Czech and Russian (up to 20%), which is unsurprising since both are morphologically richer than English.

**Agglutinative languages.** We observe different results for each language. For Finnish, character trigrams composed with bi-LSTMs achieves the best perplexity. Surprisingly, for Turkish character trigrams composed via addition is best and addition also performs quite well for other representations, potentially useful since the addition function is simpler and faster than bi-LSTMs. We suspect that this is due to the fact that Turkish morphemes are reasonably short, hence well-approximated by character trigrams. For Japanese, we improvements from character models are more modest than in other languages.

**Root and Pattern.** For these languages, character trigrams composed with bi-LSTMs also achieve the best perplexity. We had wondered whether CNNs would be more effective for root-and-pattern morphology, but since these data are unvocalized, it is more likely that non-concatenative effects are minimized, though we do

still find morphological variants with consonantal inflections that behave more like concatenation. For example, *maktab* (root: *ktb*) is written as *mktb*. We suspect this makes character trigrams quite effective since they match the tri-consonantal root patterns among words which share the same root.

**Reduplication.** For Indonesian, BPE morphs composed with bi-LSTMs model obtain the best perplexity. For Malay, the character CNN outperforms other models. However, these improvements are small compared to other languages. This likely reflects that Indonesian and Malay are only moderately inflected, where inflection involves both concatenative and non-concatenative processes.

## 5.1 Effects of Morphological Analysis

In the experiments above, we used unsupervised morphological *segmentation* as a proxy for morphological *analysis* (Table 3). However, as discussed in Section 2, this is quite approximate, so it is natural to wonder what would happen if we had the true morphological analysis. If character-level models are powerful enough to capture the effects of morphology, then they should have the predictive accuracy of a model with access to this analysis. To find out, we conducted an oracle experiment using the human-annotated morphological analyses provided in the UD datasets for Czech and Russian, the only languages in our set for which these analyses were available. In these experiments we treat the lemma and each morphological feature as a subword unit.

The results (Table 6) show that bi-LSTM composition of these representations outperforms all

Languages	Addition	bi-LSTM
Czech	51.8	<b>30.07</b>
Russian	41.82	<b>26.44</b>

Table 6: Perplexity results using hand-annotated morphological analyses (cf. Table 5).

other models for both languages. These results demonstrate that neither character representations nor unsupervised segmentation is a perfect replacement for manual morphological analysis, at least in terms of predictive accuracy. In light of character-level results, they imply that current unsupervised morphological analyzers are poor substitutes for real morphological analysis.

However, we can obtain much more unannotated than annotated data, and we might guess that the character-level models would outperform those based on morphological analyses if trained on larger data. To test this, we ran experiments that varied the training data size on three representation models: word, character-trigram bi-LSTM, and character CNN. Since we want to see how much training data is needed to reach perplexity obtained using annotated data, we use the same output vocabulary derived from the original training. While this makes it possible to compare perplexities across models, it is unfavorable to the models trained on larger data, which may focus on other words. This is a limitation of our experimental setup, but does allow us to draw some tentative conclusions. As shown in Table 7, a character-level model trained on an order of magnitude more data still does not match the predictive accuracy of a model with access to morphological analysis.

## 5.2 Automatic Morphological Analysis

The oracle experiments show promising results if we have annotated data. But these annotations are expensive, so we also investigated the use of automatic morphological analysis. We obtained analyses for Arabic with the MADAMIRA (Pasha et al., 2014).<sup>4</sup> As in the experiment using annotations, we treated each morphological feature as a subword unit. The resulting perplexities of **71.94** and **42.85** for addition and bi-LSTMs, respectively, are worse than those obtained with character trigrams (**39.87**), though it approaches the best perplexities.

<sup>4</sup>We only experimented with Arabic since MADAMIRA disambiguates words in contexts; most other analyzers we found did not do this, and would require additional work to add disambiguation.

#tokens	word	char trigram bi-LSTM	char CNN
1M	39.69	32.34	35.15
2M	37.59	36.44	35.58
3M	36.71	35.60	35.75
4M	35.89	32.68	35.93
5M	35.20	34.80	37.02
10M	35.60	35.82	39.09

Table 7: Perplexity results on the Czech development data, varying training data size. Perplexity using ~1M tokens annotated data is **28.83**.

## 5.3 Targeted Perplexity Results

A difficulty in interpreting the results of Table 5 with respect to specific morphological processes is that perplexity is measured for all words. But these processes do not apply to all words, so it may be that the effects of specific morphological processes are washed out. To get a clearer picture, we measured perplexity for only specific subsets of words in our test data: specifically, given target word  $w_i$ , we measure perplexity of word  $w_{i+1}$ . In other words, we analyze the perplexities *when the inflected words of interest are in the most recent history*, exploiting the recency bias of our LSTM-LM. This is the perplexity most likely to be strongly affected by different representations, since we do not vary representations of the predicted word itself.

We look at several cases: nouns and verbs in Czech and Russian, where word classes can be identified from annotations, and reduplication in Indonesian, which we can identify mostly automatically. For each analysis, we also distinguish between *frequent* cases, where the inflected word occurs more than ten times in the training data, and *rare* cases, where it occurs fewer than ten times. We compare only bi-LSTM models.

For Czech and Russian, we again use the UD annotation to identify words of interest. The results (Table 8), show that manual morphological analysis uniformly outperforms other subword models, with an especially strong effect for Czech nouns, suggesting that other models do not capture useful predictive properties of a morphological analysis. We do however note that character trigrams achieve low perplexities in most cases, similar to overall results (Table 5). We also observe that the subword models are more effective for rare words.

Inflection	Model	all	frequent	rare
Czech nouns	word	61.21	56.84	72.96
	characters	51.01	47.94	59.01
	char-trigrams	<u>50.34</u>	48.05	<u>56.13</u>
	BPE	53.38	49.96	62.81
	morph. analysis	<b>40.86</b>	<b>40.08</b>	<b>42.64</b>
Czech verbs	word	81.37	74.29	99.40
	characters	70.75	68.07	77.11
	char-trigrams	65.77	63.71	70.58
	BPE	74.18	72.45	78.25
	morph. analysis	<b>59.48</b>	<b>58.56</b>	<b>61.78</b>
Russian nouns	word	45.11	41.88	48.26
	characters	37.90	37.52	<u>38.25</u>
	char-trigrams	<u>36.32</u>	<u>34.19</u>	38.40
	BPE	43.57	43.67	43.47
	morph. analysis	<b>31.38</b>	<b>31.30</b>	<b>31.50</b>
Russian verbs	word	56.45	47.65	69.46
	characters	45.00	40.86	50.60
	char-trigrams	<u>42.55</u>	<u>39.05</u>	<u>47.17</u>
	BPE	54.58	47.81	64.12
	morph. analysis	<b>41.31</b>	<b>39.8</b>	<b>43.18</b>

Table 8: Average perplexities of words that occur after nouns and verbs. Frequent words occur more than ten times in the training data; rare words occur fewer times than this. The best perplexity is in **bold** while the second best is underlined.

For Indonesian, we exploit the fact that the hyphen symbol ‘-’ typically separates the first and second occurrence of a reduplicated morpheme, as in the examples of Section 2. We use the presence of word tokens containing hyphens to estimate the percentage of those exhibiting reduplication. As shown in Table 9, the numbers are quite low.

Table 10 shows results for reduplication. In contrast with the overall results, the BPE bi-LSTM model has the worst perplexities, while character bi-LSTM has the best, suggesting that these models are more effective for reduplication.

Looking more closely at BPE segmentation of reduplicated words, we found that only 6 of 252 reduplicated words have a correct word segmentation, with the reduplicated morpheme often combining differently with the notional start-of-word or hyphen character. On the other hand BPE correctly learns 8 out of 9 Indonesian prefixes and 4 out of 7 Indonesian suffixes.<sup>5</sup> This analysis supports our intuition that the improvement from BPE might come from its modeling of concatenative morphology.

## 5.4 Qualitative Analysis

Table 11 presents nearest neighbors under cosine similarity for in-vocabulary, rare, and out-of-

<sup>5</sup>We use Indonesian affixes listed in Larasati et al. (2011)

Language	type-level (%)	token-level (%)
Indonesian	1.10	2.60
Malay	1.29	2.89

Table 9: Percentage of full reduplication on the type and token level.

Model	all	frequent	rare
word	101.71	91.71	156.98
characters	<b>99.21</b>	<b>91.35</b>	<b>137.42</b>
BPE	117.2	108.86	156.81

Table 10: Average perplexities of words that occur after reduplicated words in the test set.

vocabulary (OOV) words.<sup>6</sup> For frequent words, standard word embeddings are clearly superior for lexical meaning. Character and morph representations tend to find words that are orthographically similar, suggesting that they are better at modeling dependent than root morphemes. The same pattern holds for rare and OOV words. We suspect that the subword models outperform words on language modeling because they exploit affixes to signal word class. We also noticed similar patterns in Japanese.

We analyze reduplication by querying reduplicated words to find their nearest neighbors using the BPE bi-LSTM model. If the model were sensitive to reduplication, we would expect to see morphological variants of the query word among its nearest neighbors. However, from Table 12, this is not so. With the partially reduplicated query *berlembah-lembah*, we do not find the lemma *lembah*.

## 6 Conclusion

We presented a systematic comparison of word representation models with different levels of morphological awareness, across languages with different morphological typologies. Our results confirm previous findings that character-level models are effective for many languages, but these models do not match the predictive accuracy of model with explicit knowledge of morphology, even after we increase the training data size by ten times. Moreover, our qualitative analysis suggests that they learn orthographic similarity of affixes, and lose the meaning of root morphemes.

Although morphological analyses are available

<sup>6</sup><https://radimrehurek.com/gensim/>

Model	Frequent Words			Rare Words		OOV words	
	<i>man</i>	<i>including</i>	<i>relatively</i>	<i>unconditional</i>	<i>hydroplane</i>	<i>uploading</i>	<i>foodism</i>
word	person anyone children men	like featuring include includes	extremely making very quite	nazi fairly joints supreme	molybdenum your imperial intervene	- - - -	- - - -
BPE LSTM	ii hill text netherlands	called involve like creating	newly never essentially least	unintentional ungenerous unanimous unpalatable	emphasize heartbeat hybridized unplatable	upbeat uprising handling hand-colored	vigilantism pyrethrum pausanias footway
char- trigrams LSTM	mak vill cow maga	include includes undermining under	resolutely regeneratively reproductively commonly	unconstitutional constitutional unimolecular medicinal	selenocysteine guerrillas scrofula seleucia	drifted affected conflicted convicted	tuaregs quft subjectivism tune-up
char- LSTM	mayr many mary may	inclusion insularity includes include	relates replicate relativity gravestones	undamaged unmyelinated unconditionally uncoordinated	hydrolyzed hydraulics hysterotomy hydraulic	musagte mutualism mutualists meursault	formulas formally fecal foreland
char- CNN	mtn mann jan nun	include includes excluding included	legislatively lovely creatively negatively	unconventional unintentional unconstitutional untraditional	hydroxyproline hydrate hydrangea hyena	unloading loading upgrading upholding	fordism dadaism popism endemism

Table 11: Nearest neighbours of semantically and syntactically similar words.

Query	Top nearest neighbours
kota-kota (cities)	wilayah-wilayah ( <i>areas</i> ), pulau-pulau ( <i>islands</i> ), negara-negara ( <i>countries</i> ), bahasa-bahasa ( <i>languages</i> ), koloni-koloni ( <i>colonies</i> )
berlembah-lembah (have many valleys)	berargumentasi ( <i>argue</i> ), bercakap-cakap ( <i>converse</i> ), berkemauan ( <i>will</i> ), berimplikasi ( <i>imply</i> ), berketebalan ( <i>have a thickness</i> )

Table 12: Nearest neighbours of Indonesian reduplicated words in the BPE bi-LSTM model.

in limited quantities, our results suggest that there might be utility in semi-supervised learning from partially annotated data. Across languages with different typologies, our experiments show that the subword unit models are most effective on agglutinative languages. However, these results do not generalize to all languages, since factors such as morphology and orthography affect the utility of these representations. We plan to explore these effects in future work.

## Acknowledgments

Clara Vania is supported by the Indonesian Endowment Fund for Education (LPDP), the Centre for Doctoral Training in Data Science, funded by the UK EPSRC (grant EP/L016427/1), and the University of Edinburgh. We thank Sameer Bansal, Toms Bergmanis, Marco Damonte, Federico Fancellu, Sorcha Gilroy, Sharon Goldwater, Frank Keller, Mirella Lapata, Felicia Liu, Jonathan Mallinson, Joana Ribeiro, Naomi Saphra, Ida Szubert, and the anonymous reviewers for helpful discussion of this work and comments on previous drafts of the paper.

## References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. [TensorFlow: Large-scale machine learning on heterogeneous systems](https://www.tensorflow.org/). Software available from tensorflow.org. <https://www.tensorflow.org/>.
- Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. [Polyglot: Distributed word representations for multilingual nlp](https://www.aclweb.org/anthology/W13-3520). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Sofia, Bulgaria, pages 183–192. [http://www.aclweb.org/anthology/W13-3520](https://www.aclweb.org/anthology/W13-3520).
- Emily M. Bender. 2013. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*. Morgan & Claypool Publishers.

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. [Enriching word vectors with subword information](https://arxiv.org/abs/1607.04606). *CoRR* abs/1607.04606. <http://arxiv.org/abs/1607.04606>.
- Jan A. Botha and Phil Blunsom. 2014. [Compositional Morphology for Word Representations and Language Modeling](http://jmlr.org/proceedings/papers/v32/botha14.pdf). In *Proceedings of the 31st International Conference on Machine Learning (ICML)*. Beijing, China. <http://jmlr.org/proceedings/papers/v32/botha14.pdf>.
- Danqi Chen and Christopher Manning. 2014. [A fast and accurate dependency parser using neural networks](http://www.aclweb.org/anthology/D14-1082). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 740–750. <http://www.aclweb.org/anthology/D14-1082>.
- 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](http://www.aclweb.org/anthology/D14-1179). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. <http://www.aclweb.org/anthology/D14-1179>.
- Ryan Cotterell and Hinrich Schütze. 2015. [Morphological word-embeddings](http://www.aclweb.org/anthology/N15-1140). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1287–1292. <http://www.aclweb.org/anthology/N15-1140>.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. [Transition-based dependency parsing with stack long short-term memory](http://www.aclweb.org/anthology/P15-1033). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 334–343. <http://www.aclweb.org/anthology/P15-1033>.
- Philip Gage. 1994. [A new algorithm for data compression](http://dl.acm.org/citation.cfm?id=177910.177914). *C Users J.* 12(2):23–38. <http://dl.acm.org/citation.cfm?id=177910.177914>.
- Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. [Multilingual language processing from bytes](http://www.aclweb.org/anthology/N16-1155). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 1296–1306. <http://www.aclweb.org/anthology/N16-1155>.
- Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. [Bidirectional lstm networks for improved phoneme classification and recognition](http://dl.acm.org/citation.cfm?id=1986079.1986220). In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*. Springer-Verlag, Berlin, Heidelberg, ICANN’05, pages 799–804. <http://dl.acm.org/citation.cfm?id=1986079.1986220>.
- Martin Haspelmath. 2010. *Understanding Morphology*. Understanding Language Series. Arnold, London, second edition.
- Georg Heigold, Guenter Neumann, and Josef van Genabith. 2017. [An extensive empirical evaluation of character-based morphological tagging for 14 languages](http://aclweb.org/anthology/E17-1048). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, pages 505–513. <http://aclweb.org/anthology/E17-1048>.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](https://doi.org/10.1162/neco.1997.9.8.1735). *Neural Comput.* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Katharina Kann and Hinrich Schütze. 2016. [Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology](https://doi.org/10.18653/v1/W16-2010), Association for Computational Linguistics, chapter MED: The LMU System for the SIGMORPHON 2016 Shared Task on Morphological Reinflection, pages 62–70. <https://doi.org/10.18653/v1/W16-2010>.
- Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. [Character-aware neural language models](https://arxiv.org/abs/1610.03017). In *Proceedings of the 2016 Conference on Artificial Intelligence (AAAI)*.
- Septina Dian Larasati, Vladislav Kuboň, and Daniel Zeman. 2011. *Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 119–129. [https://doi.org/10.1007/978-3-642-23138-4\\_8](https://doi.org/10.1007/978-3-642-23138-4_8).
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. [Fully character-level neural machine translation without explicit segmentation](https://arxiv.org/abs/1610.03017). *CoRR* abs/1610.03017. <http://arxiv.org/abs/1610.03017>.
- Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fernandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. [Finding function in form: Compositional character models for open vocabulary word representation](http://aclweb.org/anthology/D15-1176). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1520–1530. <http://aclweb.org/anthology/D15-1176>.
- Thang Luong, Richard Socher, and Christopher Manning. 2013. [Better word representations with recursive neural networks for morphology](https://arxiv.org/abs/1306.2580). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Association for

- Computational Linguistics, Sofia, Bulgaria, pages 104–113. <http://www.aclweb.org/anthology/W13-3512>.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. [Building a large annotated corpus of english: The penn treebank](http://www.aclweb.org/anthology/P13-1045). *Comput. Linguist.* 19(2):313–330. <http://dl.acm.org/citation.cfm?id=972470.972475>.
- Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. [Recurrent neural network based language model](http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html). In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*. International Speech Communication Association, volume 2010, pages 1045–1048. [http://www.isca-speech.org/archive/interspeech\\_2010/i10\\_1045.html](http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html).
- Yasumasa Miyamoto and Kyunghyun Cho. 2016. [Gated word-character recurrent language model](https://aclweb.org/anthology/D16-1209). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1992–1997. <https://aclweb.org/anthology/D16-1209>.
- G. David Morley. 2000. *Syntax in Functional Grammar: An Introduction to Lexicogrammar in Systemic Linguistics*. Continuum.
- Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Ctilina Mrnduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Ceneľ-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2015. [Universal dependencies 1.2 LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague](http://www.aclweb.org/anthology/P15-1162). <http://hdl.handle.net/11234/1-1548>.
- Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholi, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, pages 1094–1101. ACL Anthology Identifier: L14-1479.
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. [Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss](http://anthology.aclweb.org/P16-2067). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 412–418. <http://anthology.aclweb.org/P16-2067>.
- Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. [Co-learning of word representations and morpheme representations](http://www.aclweb.org/anthology/C14-1015). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 141–150. <http://www.aclweb.org/anthology/C14-1015>.
- Marek Rei, Gamal Crichton, and Sampo Pyysalo. 2016. [Attending to characters in neural sequence labeling models](http://aclweb.org/anthology/C16-1030). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 309–318. <http://aclweb.org/anthology/C16-1030>.
- Brian Roark and Richard Sproat. 2007. *Computational Approach to Morphology and Syntax*. Oxford University Press.
- Cicero Dos Santos and Bianca Zadrozny. 2014. [Learning character-level representations for part-of-speech tagging](http://proceedings.mlr.press/v32/santos14.html). In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*. PMLR, Beijing, China, volume 32 of *Proceedings of Machine Learning Research*, pages 1818–1826. <http://proceedings.mlr.press/v32/santos14.html>.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](http://www.aclweb.org/anthology/P16-1162). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. <http://www.aclweb.org/anthology/P16-1162>.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. *Morfessor 2.0: Toolkit for statistical morphological segmentation*. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, pages 21–24. <http://www.aclweb.org/anthology/E14-2006>.

Henning Sperr, Jan Niehues, and Alex Waibel. 2013. *Letter n-gram-based input encoding for continuous space language models*. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Sofia, Bulgaria, pages 30–39. <http://www.aclweb.org/anthology/W13-3204>.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2016. *Word representation models for morphologically rich languages in neural machine translation*. *CoRR* abs/1606.04217. <http://arxiv.org/abs/1606.04217>.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. *Charagram: Embedding words and sentences via character n-grams*. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1504–1515. <https://aclweb.org/anthology/D16-1157>.