Word Representation Models for Morphologically Rich Languages in Neural Machine Translation

Ekaterina Vylomova and **Trevor Cohn** and **Xuanli He**
The University of Melbourne
Melbourne, VIC, Australia
evylomova@gmail.com trevor.cohn@unimelb.edu.au
xuanlih@student.unimelb.edu.au

Gholamreza Haffari
Monash University
Clayton, VIC, Australia
gholamreza.haffari@monash.edu

Abstract

Out-of-vocabulary words present a great challenge for Machine Translation. Recently various character-level compositional models were proposed to address this issue. In current research we incorporate two most popular neural architectures, namely LSTM and CNN, into hard- and soft-attentional models of translation for character-level representation of the source. We propose semantic and morphological intrinsic evaluation of encoder-level representations. Our analysis of the learned representations reveals that character-based LSTM seems to be better at capturing morphological aspects compared to character-based CNN. We also show that a hard-attentional model provides better character-level representations compared to standard 'soft' attention.

1 Introduction

Models of end-to-end machine translation based on neural networks can produce excellent translations, rivalling or surpassing traditional statistical machine translation systems (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). A central challenge in neural MT is handling rare and uncommon words. Conventional neural MT models use a fixed modest-size vocabulary, such that the identity of rare words are lost, which makes their translation exceedingly difficult. Accordingly, sentences containing rare words tend to be translated much more poorly than those containing only common words (Sutskever et al., 2014; Bahdanau et al., 2015). The rare word problem is exacerbated when translating from morphologically rich languages, where the several morphological variants of words result in a huge vocabulary with a heavy tail. For example in Russian, there are at least 70 word forms for dog, encoding case, gender, age, number, sentiment and other semantic connotations. Many of them share a common lemma, and contain regular morphological affixation; consequently much of the information required for translation is present, but not in an accessible form for models of neural MT.

In many cases the OOV problem is addressed by incorporating character-level word representations largely belonging to one of two classes, namely convolutional neural networks (CNNs) and recurrent neural networks based on long-short term memory (LSTM) units (Hochreiter and Schmidhuber, 1997). But there was no investigation of what each of the models captures and how well they can model morphology in particular. In this paper, we fill this gap by evaluating of encoder-level representations of OOV words. To get the representations, we incorporate LSTM and CNN word representation models into two types of attentional machine translation models. Our evaluation includes both intrinsic and extrinsic metrics, where we compare these approaches based on their translation performance as well as their ability to recover synonyms for the rare words. Intrinsic analysis shows that there is only minor differences in end translation performance, although detailed analysis shows that character-based LSTM is overally best at capturing morphological regularities.

Proceedings of the First Workshop on Subword and Character Level Models in NLP, pages 103–108,
Copenhagen, Denmark, September 7, 2017.

2 Related Work

Most neural models for NLP rely on words as their basic units, and consequently face the problem of how to handle tokens in the test set that are out-of-vocabulary (OOV). Often these words are assigned a special UNK token, which comes at the expense of modelling accuracy. One solution to OOV problem is modelling sub-word units, using a model of a word from its composite morphemes. Luong et al. (2013) proposed a recursive combination of morphs using affine transformation, however this is unable to differentiate between the compositional and non-compositional cases. Botha and Blunsom (2014) tackle this problem by forming word representations from adding a sum of each word's morpheme embeddings to its word embedding. Morpheme based methods rely on good morphological analysers, however these are only available for a limited set of languages. Unsupervised analysers (Creutz and Lagus, 2007) are prone to segmentation errors, particularly on fusional or polysynthetic languages. In these settings, character-level word representations may be more appropriate.

Several authors have proposed convolutional neural networks over character sequences, as part of models of part of speech tagging (Santos and Zadrozny, 2014), named entity recognition (Ma and Hovy, 2016; Chiu and Nichols, 2015), language (Kim et al., 2015) and machine translation (Costa-jussà and Fonollosa, 2016; Belinkov et al., 2017). The latter one presents an in-depth analysis of representations learned by neural MT models. Another strand of research has looked at recurrent architectures, using long-short term memory units (Ling et al., 2015; Ballesteros et al., 2015) which can capture long orthographic patterns in the character sequence, as well as non-compositionality. (Lample et al., 2016) shows that incorporating biLSTM character-level word representations improves accuracy in named entity recognition task.

All of the aforementioned models were shown to either perform similar or even outperform standard word-embedding approaches. With a few notable exceptions (Vania and Lopez, 2017; Heigold et al., 2017), there was no systematic investigation of the various modelling architectures. In our work we address the question of what linguistic lexical aspects are best encoded in each type of architecture, and their efficacy as part of a machine translation model when translating from morphologically rich languages.

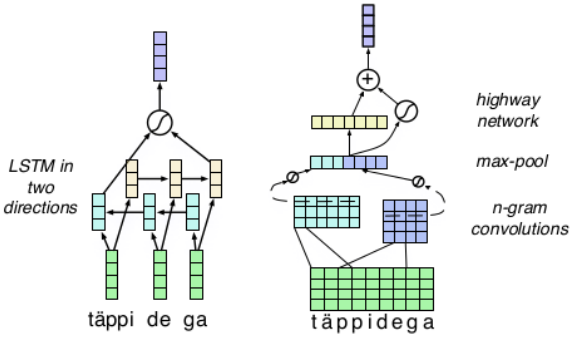

Figure 1: Model architecture for the several approaches to learning word representations, showing from left: BiLSTM over characters and the character convolution.

3 Models

Now we turn to the problem of learning word representations. We consider character level encoding methods which we compare to the baseline word embedding approach. We test two types of character representations: LSTM recurrent neural networks (RNN) and convolutional neural network (CNN).

For each type of character encoder we learn two word representations: one estimated from the characters and the word embedding.[1] Then we run max pooling over both embeddings to obtain the word representation, $\mathbf{r}_w = \boldsymbol{m}_w \oslash \boldsymbol{e}_w$, where $\boldsymbol{m}_w$ is the embedding of word w and $\boldsymbol{e}_w$ is the sub-word encoding. The max pooling operation $\oslash$ captures non-compositionality in the semantic meaning of a word relative to its sub-parts. We assume that the model would favour unit-based embeddings for rare words and word-based for more common ones.

Each word is expressed with its constituent units as follows. Let $\mathcal{U}$ be the vocabulary of sub-word units, i.e., characters, E_u be the dimensionality of unit embeddings, and $M \in \mathbf{R}^{E_u \times |\mathcal{U}|}$ be the matrix of unit embeddings. Suppose that a word w from the source dictionary is made up of a sequence of units $\mathcal{U}_w := [u_1, \ldots, u_{|w|}]$, where $|w|$ stands for the number of constituent units in the word. The resulting word representations are then fed to both attentional models as the source word embeddings.

[1] We only include word embeddings for common words; rare words share a UNK embedding.

3.1 Bidirectional LSTM Encoder

The encoding of the word is formulated using a pair of LSTMs (denoted *biLSTM*) one operating left-to-right over the input sequence and another operating right-to-left, $\boldsymbol{h}_j^{\rightarrow} = \text{LSTM}(\boldsymbol{h}_{j-1}^{\rightarrow}, \boldsymbol{m}_{u_j})$ and $\boldsymbol{h}_j^{\leftarrow} = \text{LSTM}(\boldsymbol{h}_{j+1}^{\leftarrow}, \boldsymbol{m}_{u_j})$ where $\boldsymbol{h}_j^{\rightarrow}$ and $\boldsymbol{h}_j^{\leftarrow}$ are the LSTM hidden states.[2] These are fed into perceptron with a single hidden layer and a $\tanh$ activation function to form the word representation, $\boldsymbol{e}_w = \text{MLP}\left(\boldsymbol{h}_{|\mathcal{U}_w|}^{\rightarrow}, \boldsymbol{h}_1^{\leftarrow}\right)$.

3.2 Convolutional Encoder

Another word encoder we consider is a convolutional neural network, inspired by a similar approach in language modelling (Kim et al., 2016). Let $U_w \in \mathbb{R}^{E_u \times |\mathcal{U}|_w}$ denote the unit-level representation of w, where the jth column corresponds to the unit embedding of u_j. The idea of unit-level CNN is to apply a *kernel* $\mathbf{Q}_l \in \mathbb{R}^{E_u \times k_l}$ with the width k_l to U_w to obtain a feature map $\mathbf{f_l} \in \mathbb{R}^{|\mathcal{U}|_w - k_l + 1}$. More formally, for the jth element of the feature map the convolutional representation is

$$\mathbf{f_l}(j) = \tanh(\langle U_{w,j}, \mathbf{Q}_l \rangle + b)$$

where $U_{w,j} \in \mathbb{R}^{E_u \times k_l}$ is a slice from U_w which spans the representations of the jth unit and its preceding $k_l - 1$ units, and

$$\langle A, B \rangle = \sum_{i,j} A_{ij} B_{ij} = \text{Tr}\left(AB^T\right)$$

denotes the Frobenius inner product. For example, suppose that the input has size $[4 \times 9]$, and a kernel has size $[4 \times 3]$ with a sliding step being 1. Then, we obtain a $[1 \times 7]$ feature map. This process implements a character n-gram, where n is equal to the width of the filter. The word representation is then derived by max pooling the feature maps of the kernels:

$$\forall l: \quad \mathbf{r}_w(l) = \max_j \mathbf{f_l}(j)$$

In order to capture interactions between the character n-grams obtained by the filters, a *highway network* (Srivastava et al., 2015) is applied after the max pooling layer,

$$\boldsymbol{e}_w = \boldsymbol{t} \odot \text{MLP}(\boldsymbol{r}_w) + (1 - \boldsymbol{t}) \odot \boldsymbol{r}_w,$$

where $t = \text{MLP}_\sigma(\boldsymbol{r}_w)$ is a sigmoid gating function which modulates between a $\tanh$ MLP transformation of the input (left component) and preserving the input as is (right component).

[2] The memory cells are computed as part of the recurrence, suppressed here for clarity.

Language	Ru-En	Et-En
Phrase-based Baseline	15.02	24.40
AM BILSTM$_{\text{char}}$	16.01	26.34
OSM BILSTM$_{\text{char}}$	15.81	26.14
AM CNN$_{\text{char}}$	15.90	26.14
OSM CNN$_{\text{char}}$	15.94	25.97
AM BILSTM$_{\text{word}}$	15.93	26.33
OSM BILSTM$_{\text{word}}$	15.70	26.03

Table 2: BLEU scores for re-ranking the test sets.

4 Experiments

Datasets. We use parallel bilingual data from Europarl for Estonian-English (Koehn, 2005), and web-crawled parallel data for Russian-English (Antonova and Misyurev, 2011). For preprocessing, we tokenize, lower-case, and filter out sentences longer than 30 words. We apply a frequency threshold of 5, replacing low-frequency words with a special UNK token. Table 1 presents the corpus statistics.

4.1 Extrinsic Evaluation: MT

We apply the character level models in the encoder of the neural attentional (Bahdanau et al., 2015) (AM, soft-attentional) and neural operation sequence (Vylomova et al., 2016) (OSM, hard-attentional) models, replacing the source word embedding component with a BiLSTM or CNN over characters. To evaluate translations, we re-ranked moses[3] 100-best output translations using the attentional models. The re-ranker includes standard features from moses plus an extra feature(s) for each of the models. For the AM we supply the log probability of the candidate translation, and for the OSM we add two extra features corresponding to the generated alignment and the translation probabilities. The weights of the re-ranker are then trained using MERT (Och, 2003) with 100 restarts to optimise BLEU.

Table 2 presents BLEU score results. As seen, re-ranking based on neural models' scores outperforms the phrase-based baseline. However, the translation quality of the neural models are not significantly different. We assume that this is due to re-ranking of moses translations rather than decoding. Also note that here we do not address the problem of OOV on the decoding side.

4.2 Intrinsic Evaluation

We now take a closer look at the embeddings learned by the models, based on how well they

[3] https://github.com/moses-smt.

Set	Train		Development		Test		
	tokens	types	tokens	types	tokens	types	OOV rate
Ru-En	1,639K-1,809K	145K-65K	150K-168K	35K-18K	150K-167K	35K-18K	45%
Et-En	1,411K-1,857K	90K-25K	141K-188K	21K-9K	142K-189K	21K-8K	45%

Table 1: Corpus statistics for parallel data between Russian/Estonian and English. The OOV rate are the fraction of word types in the source language that are in the test set but are below the frequency cut-off or unseen in training.

capture the *semantic* and *morphological* information in the nearest neighbour words. Learning representations for low frequency words is harder than that for high-frequency words, since low frequency words cannot capitalise as reliably on their contexts. Therefore, we split the test lexicon into 6 parts according to their frequency in the training set. Since we set out word frequency threshold to 5 for the training set, all words appearing in the lowest frequency band [0,4] are OOVs for the test set. For each word of the test set, we take its top-20 nearest neighbours from the whole training lexicon using cosine similarity.

Semantic Evaluation. We investigate how well the nearest neighbours are interchangable with a query word in the translation process. So we formalise the notion of semantics of the source words based on their translations in the target language. We use *pivoting* to define the probability of a candidate word e' to be the synonym of the query word e, $p(e'|e) = \sum_f p(f|e)p(e'|f)$, where f is a target language word, and the translation probabilities inside the summation are estimated using a word-based translation model trained on the entire initial bilingual corpora. We then take the top-5 most probable words as the gold synonyms for each query word of the test set.[4]

We measure the quality of predicted nearest neighbours using the multi-label accuracy[5], $\frac{1}{|S|}\sum_{w\in S} \mathbf{1}_{[\mathrm{G}(w)\cap \mathrm{N}(w)\neq\varnothing]}$ where $G(w)$ and $N(w)$ are the sets of gold standard synonyms and nearest neighbors for w respectively; the function $\mathbf{1}_{[C]}$ is one if the condition C is true, and zero otherwise. In other words, it is the fraction of words in S whose nearest neighbours and gold standard synonyms have non-empty overlap.

Table 3 presents the semantic evaluation results. As seen, for the *vanilla* (soft) attentional model word- and character-level representations perform quite similar. In case of the *hard* attentional model we OSM CNN_{char} outperforms other representations by a large margin.

Model Freq.	0-4	5-9	10-14	15-19	20-50	50+
Russian						
AM $\text{BILSTM}_{\text{word}}$	-	0.32	0.52	0.65	**0.81**	**0.95**
OSM $\text{BILSTM}_{\text{word}}$	-	0.36	0.49	0.61	0.76	0.91
AM $\text{BILSTM}_{\text{char}}$	0.21	0.33	0.49	0.58	0.71	0.85
OSM $\text{BILSTM}_{\text{char}}$	0.16	0.34	0.48	0.59	0.71	0.85
AM CNN_{char}	0.13	0.23	0.38	0.47	0.61	0.84
OSM CNN_{char}	**0.43**	**0.71**	**0.77**	**0.77**	**0.81**	0.81
Estonian						
AM $\text{BILSTM}_{\text{word}}$	-	0.39	0.53	0.63	0.72	0.88
OSM $\text{BILSTM}_{\text{word}}$	-	0.48	0.62	0.70	**0.79**	**0.90**
AM $\text{BILSTM}_{\text{char}}$	0.12	0.30	0.37	0.45	0.52	0.70
OSM $\text{BILSTM}_{\text{char}}$	0.13	0.39	0.48	0.55	0.63	0.78
AM CNN_{char}	0.12	0.25	0.33	0.42	0.52	0.75
OSM CNN_{char}	**0.48**	**0.70**	**0.75**	**0.76**	0.78	0.78

Table 3: Semantic evaluation of nearest neighbours using multi-label accuracy on words in different frequency bands.

Morphological Evaluation. We now turn to evaluating the morphological component. We only focus on Russian since it has a notoriously hard morphology. We run another morphological analyser, *mystem* (Segalovich, 2003), to generate *linguistically tagged* morphological analyses for a word, e.g. POS tags, case, person, plurality, etc. We represent each morphological analysis with a bit vector, where each 1 bit indicates the presence of a specific grammatical feature. Each word is then assigned a set of bit vectors corresponding to the set of its morphological analyses. As the *morphology similarity* between two words, we take the minimum of Hamming similarity[6] between the corresponding two sets of bit vectors. Table 4(a) shows the average morphology similarity between the words and their nearest neighbours across the frequency bands. Likewise, we represent the words based on their lemma features; Table 4(b) shows the average lemma similarity.

Table 5 lists top five nearest neighbours for OOV words produced by the OSM models. BiLSTMs better capture morphological similarities expressed in suffixes and prefixes. We assume this

[4]We remove query words whose frequency is less than a threshold in the initial bilingual corpora, since pivoting may not result in high quality synonyms for such words.

[5]We evaluated using mean reciprocal rank (MRR) measure as well, and obtained results consistent with the multi-label accuracy (omitted due to space constraints).

[6]The Hamming similarity is the number of bits having the same value in two given bit vectors.

Ras+po+*lag*+a+ušč+ej

Disposing (inpraes,dat,sg,partcp,plen,f,ipf,intr)

OSM CNN_{char}	OSM $BILSTM_{char}$
ras+po+lag+a+ušč+iy *disposing (inpraes,nom,sg,partcp,plen,m,ipf,inan,intr)*	ras+slab+l+ja+ušč+ej *relaxing (inpraes,dat,sg,partcp,plen,f,ipf)*
ras+po+lag+a+ušč+im *disposing (inpraes,ins,sg,partcp,plen,m,ipf,intrn)*	so+pro+voj+d+a+ušč+ej *accompanying (inpraes,dat,sg,partcp,plen,f,ipf,tran)*
ras+po+lag+a+ušč+ie *disposing (inpraes,nom,pl,partcp,plen,ipf,intr)*	ras+slab+l+ja+ušč+uju *relaxing (inpraes,acc,sg,partcp,plen,f,ipf)*
ras+po+lag+a+ušč+ih *disposing (inpraes,gen,pl,partcp,plen,ipf,intr)*	ras+po+lag+a+ušč+iy *disposing (inpraes,nom,sg,partcp,plen,m,ipf,inan,intr)*
ras+po+lag+a+ušč+i+e+sja *disposing (inpraes,nom,pl,partcp,plen,ipf,act)*	pro+dvig+a+ušč+ej *promoting (inpraes,dat,sg,partcp,plen,f,ipf,act)*

S+*konfigur*+ir+ova+ť

Configure (v,pf,tran,inf)

OSM CNN_{char}	OSM $BILSTM_{char}$
s+konfigur+ir+ui+te *configure (v,pf,tran,pl,imper,2p)*	konfigur+ir+ova+ť *configure (v,ipf,tran,inf)*
s+konfigur+ova+li *configured (v,pf,tran,praet,pl,indic)*	s+korrekt+ir+ova+ť *adjust (v,pf,tran,inf)*
s+konfigur+ova+n *configured (v,pf,tran,praet,sg,partcp,brev,m,pass)*	s+*koordin*+ir+ova+ť *coordinate (v,pf,tran,inf)*
s+konstru+ir+ova+ť *construct (v,pf,tran,inf)*	s+fokus+ir+ova+ť *focus (v,pf,tran,in)*
s+kompil+ir+ova+ť *compile (v,pf,tran,inf)*	s+kompil+ir+ova+ť *compile (v,pf,tran,inf)*

Table 5: Analysis of the five most similar Russian words (initial word is OOV), under the OSM CNN_{char} and OSM $BILSTM_{char}$ word encodings based on cosine similarity. The diacritic ´ indicates softness. **POS tags**: *s*-noun, *a*-adjective, *v*-verb; **Gender**: *m*-masculine, *f*-feminine, *n*-neuter; **Number**: *sg*-singular, *pl*-plural; **Case**: *nom*-nominative, *gen*-genitive, *dat*-dative, *acc*-accusative, *ins*-instrumental, *abl*-prepositional, *loc*-locative; **Tense**: *praes*-present, *inpraes*-continuous, *praet*-past, *pf*-perfect, *ipf*-imperfect; *indic*-indicative; **Transitivity**: *trans*-transitive, *intr*-intransitive; **Adjective form**: *br*-brevity, *plen*-full form, *poss*-possessive; **Comparative**: *supr*-superlative, *comp*-comparative; **Noun person**: *1p*-first, *2p*-second, *3p*-third;

Model \ Freq.	**0-4**	**5-9**	**10-14**	**15-19**	**20-50**	**50+**
AM $BILSTM_{word}$	-	0.70	0.73	075	0.78	0.82
OSM $BILSTM_{word}$	-	0.74	0.77	0.78	0.81	0.84
AM $BILSTM_{char}$	0.90	0.82	0.83	0.83	0.84	0.82
OSM $BILSTM_{char}$	**0.91**	**0.84**	**0.85**	**0.85**	**0.86**	**0.86**
AM CNN_{char}	0.82	0.76	0.77	0.78	0.79	0.81
OSM CNN_{char}	0.79	0.80	0.79	0.79	0.79	0.79

(a)

Model \ Freq.	**0-4**	**5-9**	**10-14**	**15-19**	**20-50**	**50+**
AM $BILSTM_{word}$	-	0.02	0.04	0.07	0.11	0.18
OSM $BILSTM_{word}$	-	0.03	0.05	0.06	0.09	0.15
AM $BILSTM_{char}$	0.08	0.06	0.10	0.11	0.12	0.21
OSM $BILSTM_{char}$	0.05	0.05	0.08	0.10	0.13	0.18
AM CNN_{char}	0.04	0.02	0.05	0.06	0.1	0.15
OSM CNN_{char}	**0.20**	**0.37**	**0.41**	**0.42**	**0.44**	**0.41**

(b)

Table 4: Morphology analysis for nearest neighbours based on (a) Grammar tag features, and (b) Lemma features, evaluated on Russian.

is due to the fact that they are naturally biased towards most recent inputs. CNNs, on the other hand, are more invariant of character positions and provide whole-word similarity.

5 Conclusion

We studied two types of attentional models augmented by CNN and LSTM encodings. Our experiments demonstrate that representation of out-of-vocabulary words with their sub-word units on the source side did not lead to a significant improvement in overall quality of machine translation; however LSTMs applied to character sequences are more capable at learning morphological patterns. Moreover, a hard attention mechanism leads to better capturing of semantic and morphological regularities.

References

Alexandra Antonova and Alexey Misyurev. 2011. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*. Association for Computational Linguistics, pages 136–144.

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

Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. *arXiv preprint arXiv:1508.00657* .

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural ma-

chine translation models learn about morphology? *arXiv preprint arXiv:1704.03471* .

Jan A Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. *arXiv preprint arXiv:1405.4273* .

Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308* .

Marta Costa-jussà and Jose Fonollosa. 2016. Character-based neural machine translation. *arXiv preprint arXiv:1603.00810* .

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)* 4(1):3.

Georg Heigold, G ünter Neumann, and Josef van Genabith. 2017. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. volume 1, pages 505–513.

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 *EMNLP*. pages 1700–1709.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*.

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

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*. volume 5, pages 79–86.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* .

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096* .

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*. Citeseer, pages 104–113.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354* .

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 160–167.

Cicero D. Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1818–1826.

Ilya Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In *MLMTA*. Citeseer, pages 273–280.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in Neural Information Processing Systems*. pages 2368–2376.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems (NIPS)*. Montréal, pages 3104–3112.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? *arXiv preprint arXiv:1704.08352* .

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2016. Word representation models for morphologically rich languages in neural machine translation. *arXiv preprint arXiv:1606.04217* .