Natural Language Engineering **25** (1): 147–169.
doi:10.1017/S1351324918000281

The effect of morphology in named entity recognition with sequence tagging†

ONUR GÜNGÖR

Department of Computer Engineering, Bogazici University, Istanbul
Huawei R&D Center, Istanbul, Turkey
email: onurgu@boun.edu.tr

TUNGA GÜNGÖR and SUZAN ÜSKÜDARLI

Department of Computer Engineering, Bogazici University, Istanbul, Turkey
e-mail: gungort@boun.edu.tr, suzan.uskudarli@boun.edu.tr

(*Received 9 August 2017; revised 2 July 2018; accepted 2 July 2018; first published online 27 July 2018*)

Abstract

This work proposes a sequential tagger for named entity recognition in morphologically rich languages. Several schemes for representing the morphological analysis of a word in the context of named entity recognition are examined. Word representations are formed by concatenating word and character embeddings with the morphological embeddings based on these schemes. The impact of these representations is measured by training and evaluating a sequential tagger composed of a conditional random field layer on top of a bidirectional long short-term memory layer. Experiments with Turkish, Czech, Hungarian, Finnish and Spanish produce the state-of-the-art results for all these languages, indicating that the representation of morphological information improves performance.

1 Introduction

Named entity recognition (NER) is an important task in natural language processing that aims to discover references to entities in text. The task was first introduced in the sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim 1996) as a short-term subtask. At that time, it was thought that a practical system could be developed in a relatively short time, which could also serve as a domain-independent tool for other information extraction tasks. To accomplish the NER task, portions of text were selected so that the selected text refers to the same entity in all possible contexts in which they exist, and does not refer to anything else in contexts in which that entity does not exist, thus being rigid designators (Kripke 1982).

The following examples exemplify the task and give hints about the complexity of the task. First of all, a single entity might be referred with various phrases: the terms ‘JFK’, ‘Kennedy’ or ‘John F. Kennedy’ may all refer to the same entity in

†This research was supported by Boğaziçi University Research Fund (BAP) under Grant 13083.

relevant contexts – the thirty-fifth president of the United States. Second, the phrases 'JFK' or 'John F. Kennedy' might refer to the airport in New York City. In the definition of the task made in MUC-6, the most prominent designators were person names, geographical locations and organization names, which became the simplest definition of the NER task.

The demand for NER in application domains such as of news outlets, social media and e-commerce hubs has further motivated research in this area (see Section 2). In a recent work, long short-term memory (LSTM) and gated recurrent units have been utilized for this task (Huang, Xu and Yu 2015; Lample *et al.* 2016; Ma and Hovy 2016; Yang, Salakhutdinov and Cohen 2016). However, these approaches are not well studied for morphologically rich languages (MRLs). MRLs retain information in the morphology of the surface form of words, which is present in the syntax and word n-grams in other languages. For example, in Turkish, a MRL, the word 'İstanbul'daydı' means 'he/she was in Istanbul', which embodies the tense and the locative case. This makes morphological understanding more important in comparison to languages with simpler morphological mechanisms. Thus, a considerable amount of research has been undertaken to understand the morphological properties in these languages.

The motivation behind this study can be explained using the same Turkish example. The morphological analysis of the word 'İstanbul'daydı' is 'İstanbul+Noun+Prop+A3sg+Pnon+Loc^DB+Verb+Zero+Past+A3sg'. The notation shown in this representation of morphological information originates from the widely accepted representation scheme for Turkish (Oflazer 1994). The analysis denotes that the root word 'İstanbul' is a proper noun in locative case and it is not marked with a possessive marker (see Section 3.1.1). The final word is a verb in past tense of third person singular agreement. One can suggest that the locative case tag and the fact that it is not possessed by anyone or anything might be a good indicator of being a named entity.

The hypothesis of this work is that morphological tags capture syntactic and semantic information and, thus, help in improving the NER performance. Thus, sequence tagging models that rely on LSTM networks are employed to provide evidence to this hypothesis and to solve the NER task for MRLs. One can argue that the literature already addressed this issue by proposing character-based embeddings in word representations (Lample *et al.* 2016) and entities tagged at the character level (Kuru, Can and Yuret 2016). Moreover, it is possible to note that morphological tags have been employed in the past for the NER task (Tür, Hakkani-Tür and Oflazer 2003; Yeniterzi 2011). However, our work treats the morphological analysis in a number of different ways that can be applied to many MRLs and is the first to propose an embedding-based framework for representing the morphological analysis in the context of NER.

The main contributions of this work are a state-of-the-art system for NER in MRLs and providing evidence that reveals augmenting word representations with morphological embeddings improves NER performance.

The paper is organized as follows: Section 2 summarizes related work. Section 3 provides information about MRLs and Bi-LSTM networks utilized in the proposed

model. Section 4 presents the proposed model. Section 5 presents the experiments in Turkish, Czech, Hungarian, Finnish and Spanish to evaluate our approach. In Section 5, the results of these experiments and a comparison of the proposed model's state-of-the-art results with the previous work are also given. Section 6 makes concluding remarks.

2 Related work

NER is closely related to complex natural language understanding tasks such as relation extraction (Miwa and Bansal 2016), knowledge base population (Rao, McNamee and Dredze 2013) and question answering (Lee, Hwang and Jang 2007; Liu and Ren 2011; Lee *et al.* 2017). Furthermore, NER systems are often part of search engines (Guo *et al.* 2009) and machine translation systems (Babych and Hartley 2003).

Early studies proposed compiling lists of people, place and organization names and exploiting them to decide whether there exists a named entity using hand-crafted rules (Appelt *et al.* 1995; Humphreys *et al.* 1998). Traditional approaches typically use several hand-crafted features such as capitalization, word length, gazetteer related features and syntactic features (part-of-speech (POS) tags, chunk tags, etc.). A wide range of machine learning-based methods have also been proposed to address the NER task. Some of the well-known approaches are conditional random fields (CRFs) (McCallum and Li 2003; Finkel, Grenager and Manning, 2005), maximum entropy (Borthwick 1999), bootstrapping (Jiang and Zhai 2007; Wu *et al.* 2009), latent semantic association (Guo *et al.* 2009) and decision trees (Szarvas, Farkas and Kocsor 2006). These techniques are generally used to create classification models that act on every token to decide whether there is an entity on that position of the text or not.

Recently, deep learning models have been instrumental in deciding how the parts of the input should be composed to form the most beneficial features leading to state-of-the-art results (Collobert *et al.* 2011). One of the key issues is the determination of how to represent the words. This is due to the symbolic nature of words. These methods rely on simple tokenization by white space and employ distributional hypothesis (Harris 1954). The research in this direction led to the use of fixed-length vectors in a dense space that improved the overall performance of many tasks, such as sentiment analysis (Socher *et al.* 2013), syntactic parsing (Collobert and Weston 2008), language modeling (Mikolov *et al.* 2010), POS tagging and NER (Collobert *et al.* 2011). These word representations or embeddings are automatically learned either during or before the training phase using methods such as Word2Vec (Mikolov *et al.* 2013), GloVe (Pennington, Socher and Manning 2014) and fastText (Bojanowski *et al.* 2017). The incorporation of morphology into this type of word embeddings was proposed for language modeling (Luong, Socher and Manning 2013; Santos and Zadrozny 2014; Bhatia, Guthrie and Eisenstein 2016; Lankinen *et al.* 2016; Xu and Liu 2017) and for morphological tagging and segmentation (Shen *et al.* 2016; Cotterell and Schütze 2017).

Built upon these findings, new approaches have been proposed that treat the NER task as a sequence labeling problem (Huang, Xu and Yu 2015; Lample *et al.*

2016; Ma and Hovy 2016; Yang *et al.* 2016). These studies employ LSTM or gated recurrent unit components to capture the syntactic and semantic relations between the units that make up a natural language sentence. In Huang *et al.* (2015), a Bi-LSTM network with a CRF layer on label scores is proposed. A special fixed-size representation is prepared for each word. The first component of this representation is the spelling features extracted from the surface forms such as whether all letters are lowercase or whether the first letter is uppercase. The second component is composed of a feature for each word bi-gram and tri-gram that exists in the context. The resulting representation is then fed to a word-level LSTM. This network's NER performance is reported to be comparable to the performance of the state-of-the-art studies. Another similar approach processes characters in each surface form in the sentence with a convolutional neural network (Ma and Hovy 2016). This helps the network to automatically represent the features that are extracted according to rules that are carefully designed in the previous work. The word representations formed with this additional component are then fed to a word-level LSTM resulting in state-of-the-art performance. Two different approaches use LSTMs (Lample *et al.* 2016) and gated recurrent units (Yang *et al.* 2016) instead of convolutional neural networks at the character and word levels otherwise similar to other work (Ma and Hovy 2016). These two studies report results similar to each other and improve the state-of-the-art results.

There are previous studies that give special importance to MRLs for the NER task. Tür *et al.* (2003) employed the last inflectional group in a statistical model that relies on maximum *a posteriori* estimation for Turkish. In another work, morphemes in a morphological analysis are used as features of a CRF model (Yeniterzi 2011). Şeker and Eryiğit (2012) also employed a CRF model with more extensive features and added gazetteers. Demir and Özgür (2014) used similar features for a CRF model with the addition of word embeddings. In our work, however, the disambiguated morphological analysis is used to generate a fixed-length vector in a number of different ways (Section 4.2), which is called a morphological embedding. This morphological embedding is composed with pretrained word embeddings and character-based embeddings to obtain a word representation for each word and employed in a setting similar to the previous work (Ma and Hovy 2016).

3 Background

This section provides background information about the MRLs examined in this work and the bidirectional LSTM (Bi-LSTM) model that is used for sequence tagging for NER.

3.1 Morphologically rich languages

The languages in which some of the syntactical functions of words are expressed by morphological phenomena within the surface forms of the words are called MRLs. In these languages, the number of words that can be generated from a single root word is very high in general. A significant number of inflections and derivations are

possible for most root nouns and verbs. In practice, however, this potential is not fully realized where a few of the affixes are attached to a stem in succession to form new words. Regardless, the number of words that can be obtained from the root words is very large. This expressiveness gives rise to complications in applications, such as data sparseness due to words with many alternative affixes.

The following sections describe the basic characteristics of the MRLs Turkish, Czech, Hungarian, Finnish and Spanish.

3.1.1 Turkish

Turkish is an agglutinative language, which expresses most syntactic information through the morphology of the surface form of the words. Thus, studies in Turkish natural language processing have mainly focused on the morphological analysis of words. To this end, a finite state transducer based on a two-level formalism (Koskenniemi 1983) was introduced (Oflazer 1994) to capture the rules of Turkish morphology (Underhill 1976; Lewis 1991). The notation introduced in this work is considered the standard for morphological analysis in Turkish. The morphological analysis of the word 'İstanbul'daydı' (he/she was in İstanbul) is as follows:

```
İstanbul+Noun+Prop+A3sg+Pnon+Loc^DB+Verb+Zero+Past+A3sg
```

where 'Prop' indicates a proper noun, 'A3sg' denotes the third singular person agreement and 'Pnon' signifies that no possessive agreement is active. 'DB' (derivational boundary) indicates a transition of the POS usually induced by a derivational suffix. It also marks the beginning of a new sequence of inflectional morphemes called inflectional group (Oflazer 2003). In this example, the derivation is triggered by the '-ydı' suffix, which is decoded with the 'Past' (past tense) tag.

3.1.2 Finnish

Finnish is an agglutinative language that exhibits vowel harmony and consonant gradation. Finnish uses derivational suffixes to a great extent. In our work, the morphological tagging of Finnish text is done by a tool called FinnPos (Silfverberg *et al.* 2016), which is an averaged structured perceptron classifier. FinnPos relies on Omorfi (Pirinen 2014) for morphological labels and lemmatization. In this work, the tool named `ftb-label` from the FinnPos package was used to obtain the disambiguated analysis as the morphological information associated with the word.

An example output of the system for the word 'Tampereella' (in Tampere) is as follows:

```
tampere+[POS=NOUN]+[PROPER=PROPER]+[NUM=SG]+[CASE=ADE]
```

which tags the word as a singular proper noun in adessive case.

3.1.3 Czech

Czech is a language with free word order known for its rich morphological properties. The morphological tags for Czech in our work is given in the NER dataset by Ševčíková *et al.* (2007), which is actually a subset of the Prague Dependency Treebank 2.0 (Hajič *et al.* 2006). These tags consist of fifteen character string where each position encodes a different morphological aspect. The tags in this treebank were labeled by seven annotators in two phases. In the first phase, annotators were given the output of a morphological tagger and were requested to select the best option. In the second phase, the discrepancies in annotator responses were resolved by another person (Hajič *et al.* 2017). For example, the word 'dlaně' (palms) is decoded as follows:

```
dlaň+NNFS2-----A----
```

which tags the word as a common noun, feminine, plural, genitive and not-negated. The tag can be decoded by a simple lookup in the tables provided in the morphological annotation manual for the Prague Dependency Treebank 2.0 (Hana *et al.* 2005).

3.1.4 Hungarian

Hungarian is also a MRL with free word order. In this work, the morphological features produced by the `magyarlanc` tool are associated with words (Zsibrita, Vincze and Farkas 2013). There are three main morphological coding schemes for Hungarian: Humor (Proszeky and Tihanyi 1993), MSD (Erjavec 2010) and KR (Tron *et al.* 2006). `magyarlanc` uses a harmonization of these schemes (Farkas *et al.* 2010), which is also adopted in this work.

As an example, the morphological features for 'nyelvészek' (linguists) is as follows:

```
nyelvész+Case=Nom+Number=Plur
```

which indicates the nominal case and plurality.

3.1.5 Spanish

Spanish is a Romance language that is not generally considered as a MRL. We examine it along with MRLs because it differs from many languages as the number of conjugated forms per verb can be as high as forty-seven. A simplified version of the POS tags from the AnCora treebank[1] was used to obtain the morphological

[1] See `http://clic.ub.edu/corpus/en/ancora` and `https://web.archive.org/web/20160325024315/http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html`

tags of the words. Here, tags do not specify the morphological features separately (as in the other four languages); they carry morphological information attached to the word. For instance, the Spanish word 'visitada' is labeled as 'VMP', indicating the past participle form of the verb. Here, 'V' indicates a verb, 'M' states that the verb is principal and 'P' indicates a participle verb.

For sentences within the corpus, the supplied POS tags are used, while for sentences outside the corpus, the Stanford POS tagger is employed.

3.2 Bidirectional LSTMs

Bi-LSTM is a type of sequential neural network that is composed of two LSTM modules, in which the first one reads the input sequence in the forward order and the second one reads it in the reverse order. The backwards component is important as it captures information about subsequent words, which can be highly significant in natural language processing tasks.

LSTM models were introduced to solve the vanishing gradients problem of recurrent neural networks (Hochreiter and Schmidhuber 1997). The sequential formulation of recurrent neural network is defined in terms of the following functions given a sequence of input vectors x_t of size d:

$$h_t = \tanh(Ux_t + Wh_{t-1})$$
$$o_t = \text{softmax}(Vh_t)$$

where h_t is the hidden state corresponding to item t of size p, o_t is the output vector corresponding to item t of size p, U is a $p \times d$ matrix and W, V are matrices of size $p \times p$.

LSTM differs from recurrent neural network in how it computes the output and hidden vectors. The following equations define an LSTM architecture (Hochreiter and Schmidhuber 1997):

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1})$$
$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1})$$
$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1})$$
$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1})$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ \tanh(c_t)$$

where σ is the sigmoid function and $\circ$ is the element-wise multiplication. In this architecture, variables i_t, f_t and o_t represent the will of the LSTM to *memorize* the cell's new value $\tilde{c}_t$, *forget* the contents of the previous cell's value c_{t-1} and *display* the current cell's value c_t, respectively. The information is carried by c_{t-1} and h_{t-1} from the previous item. In this architecture, the hidden state c_t is calculated as a parameterized sum of the previous hidden state c_{t-1} and $\tilde{c}_t$, which is a

Fig. 1. NER sample with IOB tagging scheme.

nonlinear function of the input x_t and the previous output h_{t-1}. This eliminates the repetitive multiplication performed in the recurrent neural network architecture, thereby solving the vanishing gradient problem.

For a single LSTM cell, the output vector h_t can be used for classification, regression or as input to upper layers of the neural network. In the case of Bi-LSTM, two LSTM cells, one for forwards and one for backwards, with their corresponding output vectors $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are defined. The concatenation $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$ is the output of a Bi-LSTM component (Graves and Schmidhuber 2005).

An LSTM cell is trained by learning the variables, the $p \times d$ matrices $W^{(i)}, W^{(f)}, W^{(o)}, W^{(c)}$ and the $p \times p$ matrices $U^{(i)}, U^{(f)}, U^{(o)}, U^{(c)}$. Learning is performed by backpropagation through time with an update rule of choice (i.e., vanilla stochastic gradient descent or a more advanced gradient-based algorithm). The training of Bi-LSTM is the same with additional parameters for the extra LSTM.

4 Model

In this work, the NER problem is treated as a sequence tagging problem where the model expects the input sentence to be labeled with its corresponding entities. Figure 1 shows a sentence that is labeled with the Inside Outside Beginning (IOB) scheme. The IOB tagging scheme labels the first token of an entity by prefixing its tag with 'B-', consecutive tokens of the same entity with 'I-' and tokens not associated with any entity simply as 'O', which is omitted in the figures for purposes of simplicity. So, the sample in the figure contains three named entities: 'UN' as an organization name, 'Rolf Ekéus' as a person name and 'Baghdad' as a location name.

The input sentence of length n is defined as $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2 \ldots, y_n)$. Each x_i is a fixed-length vector of size d, consisting of embeddings that represent the ith word. y_i is a vector of size K such that $y_{ik} = 1$ if and only if the correct tag is the kth tag in our tag vocabulary of size K, otherwise $y_{ik} = 0$. Further details of word representations are provided in Section 4.1. The words x_i are then fed to a Bi-LSTM, which is composed of two LSTMs (Hochreiter and Schmidhuber 1997) treating the input forwards and backwards, respectively (see Section 3.2).

The forward and backward components' cell matrices $\overrightarrow{H}$ and $\overleftarrow{H}$ are both of size $n \times p$, where p is the number of dimensions of one component of the Bi-LSTM. Figure 2 describes how the proposed model works with a Turkish sentence as an example. $\overrightarrow{H}_{i,j}$ denotes the value of the jth dimension of the ith output vector of the forward component that corresponds to the ith word in the sentence, whereas the corresponding value in the backward component is denoted by $\overleftarrow{H}_{n-i+1,j}$. The concatenation of rows $\overrightarrow{H}_i$ and $\overleftarrow{H}_{n-i+1}$ from $\overrightarrow{H}$ and $\overleftarrow{H}$, respectively, are fed to a

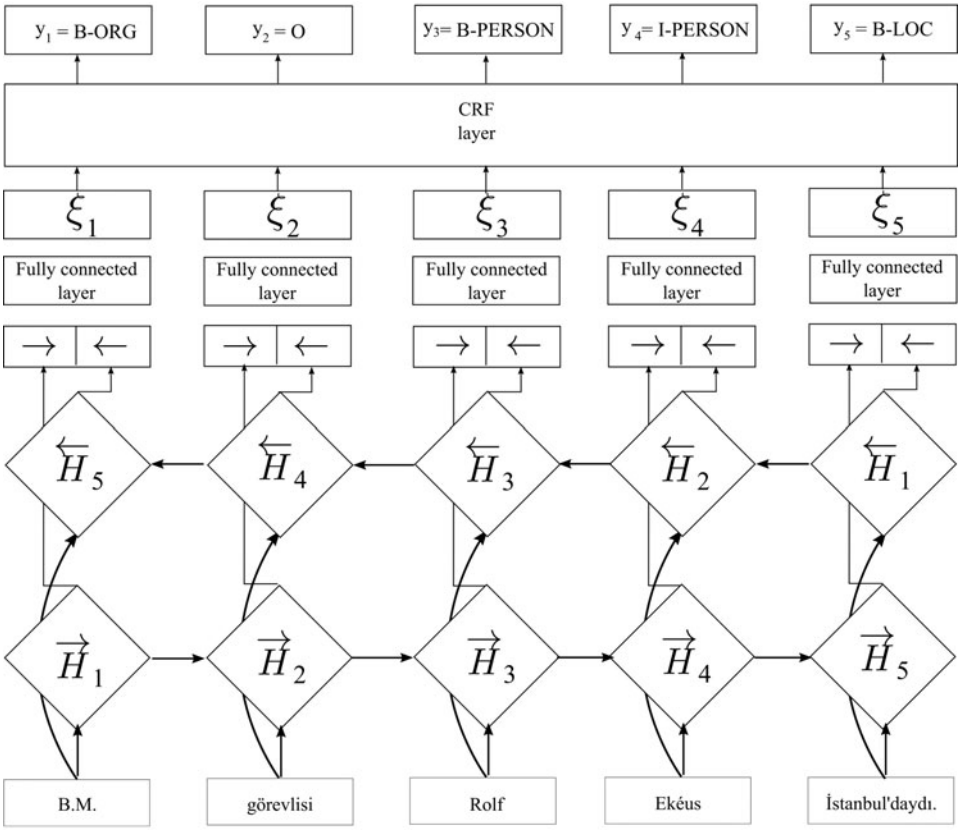

Fig. 2. Processing a Turkish sentence with the proposed model. Words are represented with a fixed-length vector that combines the word, character and morphological embeddings (see Section 4.2).

fully connected hidden layer of K output neurons for each of the n input words. The output of this fully connected layer for the ith word is denoted by ξ_i.

A CRF (Lafferty, McCallum and Pereira 2001) based approach is followed to predict the entity tags. CRF-based approaches model the dependencies between consecutive input units better than the approaches that only try to predict the correct tag based on ξ_i[2]. The advantage of the CRF model stems from the logical requirements of the IOB tagging scheme. For instance, the scheme allows tags that start with 'I-' only after a 'B-' or 'I-' tag of the same type. That is, 'I-PERSON' might only come after 'B-PERSON' or 'I-PERSON', and not after 'O' or 'B-LOCATION' or others. Moreover, the sequences in the corpus might indicate an ordering relation

[2] At this point, it is possible to exponentiate the values of ξ_i vectors and normalize them to obtain a vector which we utilize to estimate the probabilities of each tag: $\tilde{\xi}_{i,k} = \frac{\exp(\xi_{i,k})}{\sum_{\kappa=1}^{K} \exp(\xi_{i,\kappa})}$ and use $\tilde{\xi}_i$ in a cross-entropy loss function to optimize the parameters of the model: $s(x_i, y_i) = -\sum_{k=1}^{K} y_{i,k} \log(\tilde{\xi}_{i,k})$, where x_i is the ith word and y_i encodes the correct tag for the ith word. However, this approach is weaker than using a loss function that also models the dependencies between the consecutive input units.

between two tag types. For instance, 'LOCATION' tags may tend to appear more frequently before 'PERSON' tags compared to the other way around. Since CRF models allocate a probability to every valid sequence, it is possible to determine which tag sequences do not adhere to these rules as well as assigning higher probability to sequences that are in line with the ordering relations.

In order to implement a CRF model, the tag score vector ξ_i at each position i is treated as the observation score obtained from the fully connected layer and the following objective function for a sample sentence X is minimized:

$$s(X, y) = \sum_i A_{y_i, y_{i+1}} + \sum_i \xi_{i, y_i} \tag{1}$$

where $A_{i,j}$ represents the score of a transition from tag i to tag j . Then, the most probable tagging sequence y^* is as follows:

$$y^* = \underset{y'}{\operatorname{argmax}}\, s(X, y').$$

4.1 Embeddings

It has been shown that modeling units of information in a natural language input as fixed-length vectors, called embeddings, is more effective at encoding semantic properties of the words compared to using manually designed features (Turian, Ratinov and Bengio 2010; Collobert *et al.* 2011). This work utilizes embeddings where the input words, x_i, are represented as fixed-length vectors composed of three components: *word*, *character* and *morphological.*

Word embeddings: For each unique word, a vector of length d_w is defined. As mentioned above, word representations are connected to the final loss expression. This indirect relation makes them parameters of the model. Thus, it is possible to optimize each of the d_w dimensions for the target task for each distinct word. However, these parameters are not learned from scratch during training. The word embeddings are initialized to vectors obtained through approaches like skipgram with negative sampling (Mikolov *et al.* 2013) and fastText (Bojanowski *et al.* 2017). If a corpus larger than the Wikipedia for a language is available, the skipgram algorithm is employed to obtain the word embeddings using that corpus. If such a corpus is not available, we use the word embeddings that are pretrained by Bojanowski *et al.* (2016) using Wikipedia.

Character embeddings: In addition to the word embedding for a word, the covert relationships in the character sequence of the word is of value (Ma and Hovy 2016). To capture these relationships, a separate Bi-LSTM component is used for this embedding type with a cell dimension of d_c. This Bi-LSTM component is fed with the characters of the surface form of the ith word. After all the characters are processed by the Bi-LSTM component, the last cell's output of the forward and backward LSTMs are concatenated to obtain the *character* embedding of the word of length $2d_c$ (see Figure 3).

Morphological embeddings: These embeddings are constructed similar to *character* embeddings. In this case, the tags of the morphological analysis are treated as a

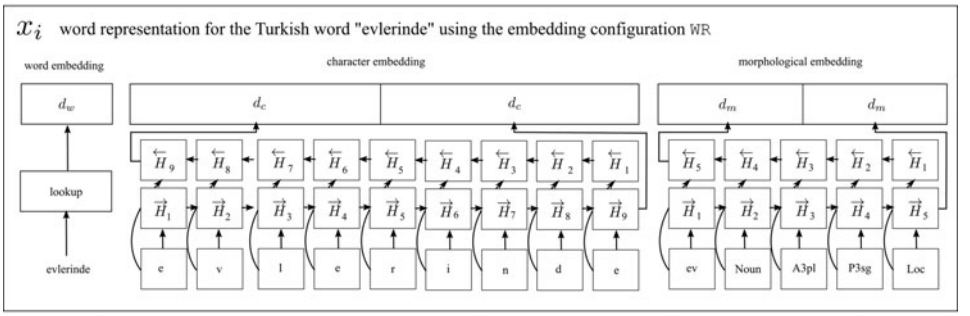

Fig. 3. The representation of the Turkish word 'evlerinde' according to the proposed model featuring all three components: word, character and morphological embeddings.

sequence and fed to a separate Bi-LSTM component for *morphological* embeddings, resulting in a vector of length $2d_m$. Several alternative combinations to serve as morphological tag sequences are described in Section 4.2.

When all of the components are active, the total size of a word representation is $d = d_w + 2d_m + 2d_c$. Figure 3 shows the representation for the Turkish word 'evlerinde' (in their houses). This representation is then fed into the sentence-level Bi-LSTM described in Section 4.[3]

4.2 Morphological embedding configurations

In order to determine an effective configuration for extracting the syntactic and semantic information in the morphological analysis of a word, experiments with a total of four different combinations of morphological tags were performed. In MRLs, it is common for a word to have more than one possible morphological analysis. The correct analysis within a context is determined using a morphological disambiguator for that language. For example, the Turkish word 'evlerinde' has three different meanings depending on the context: 'in their house', 'in their houses' and 'in his/her houses'. If the correct sense in a particular context is the last one, then the disambiguator will output the morphological analysis 'ev+Noun+A3pl+P3sg+Loc'. Here, 'A3pl' indicates third person plural, 'P3sg' is the possessive marker for third person singular and 'Loc' is the locative case marker. After the correct morphological analysis for a word is determined, the embeddings are formed as explained below.

Table 1 shows an example for each morphological embedding configuration employed in this work for each language. The first one uses the root accompanied

[3] Experiments with three separate sentence-level Bi-LSTMs, one for each of the components of our word representation were also done. However, initial experiments with this model did not give good enough results to proceed further. By doing this, it was thought that this might help in training the Bi-LSTMs so that they are better customized to their specific input embeddings. This is basically the same as the model described above. However, in separate Bi-LSTM mode, although the outputs from each of the separate Bi-LSTMs are concatenated too, each separate Bi-LSTM is fed with only one component of the word representation. For example, morphological embeddings are fed as if only they are available, while word embeddings are fed into another Bi-LSTM and the character embeddings are fed into a third Bi-LSTM.

Table 1. *Embedding configurations for Turkish, Czech, Hungarian, Finnish and Spanish:* WITH ROOT (WR), WITHOUT ROOT (WOR), WITH ROOT AND AFTER LAST DB (WR_ADB), CHAR

TR	WR	(‘ev’, ‘Noun’, ‘A3pl’, ‘P3sg’, ‘Loc’)
	WOR	(‘Noun’, ‘A3pl’, ‘P3sg’, ‘Loc’)
	WR_ADB	(‘İstanbul’, ‘Verb’, ‘Zero’, ‘Past’, ‘A3sg’)
	CHAR	(‘e’, ‘v’, ‘+’, ‘N’, ‘o’, ‘u’, ‘n’, ‘+’, ‘A’, ‘3’, ‘p’, ‘l’, ‘+’, ‘P’, ‘3’, ‘s’, ‘g’, ‘+’, ‘L’, ‘o’, ‘c’)
CS	WR	(‘prezident’, ‘NNMS1-----A----’)
	WOR	(‘NNMS1-----A----’)
	CHAR	(‘p’, ‘r’, ‘e’, ‘z’, ‘i’, ‘d’, ‘e’, ‘n’, ‘t’, ‘+’, ‘N’, ‘N’, ‘M’, ‘S’, ‘1’, ‘-’, ‘-’, ‘-’, ‘-’, ‘-’, ‘A’, ‘-’, ‘-’, ‘-’, ‘-’)
HU	WR	(‘Magyar’, ‘PROPN’, ‘Case=Nom’, ‘Number=Sing’)
	WOR	(‘PROPN’, ‘Case=Nom’, ‘Number=Sing’)
	CHAR	(‘M’, ‘a’, ‘g’, ‘y’, ‘a’, ‘r’, ‘+’, ‘P’, ‘R’, ‘O’, ‘P’, ‘N’, ‘+’, ‘C’, ‘a’, ‘s’, ‘e’, ‘=’, ‘N’, ‘o’, ‘m’, ‘+’, ‘N’, ‘u’, ‘m’, ‘b’, ‘e’, ‘r’, ‘=’, ‘S’, ‘i’, ‘n’, ‘g’)
FI	WR	(‘tampere’, ‘[POS=NOUN]’, ‘[PROPER=PROPER]’, ‘[NUM=SG]’, ‘[CASE=ADE]’)
	WOR	(‘[POS=NOUN]’, ‘[PROPER=PROPER]’, ‘[NUM=SG]’, ‘[CASE=ADE]’)
	CHAR	(‘t’, ‘a’, ‘m’, ‘p’, ‘e’, ‘r’, ‘e’, ‘+’, ‘[’, ‘P’, ‘O’, ‘S’, ‘=’, ‘N’, ‘O’, ‘U’, ‘N’, ‘]’, ‘+’, ‘[’, ‘P’, ‘R’, ‘O’, ‘P’, ‘E’, ‘R’, ‘=’, ‘P’, ‘R’, ‘O’, ‘P’, ‘E’, ‘R’, ‘]’, ‘+’, ‘[’, ‘N’, ‘U’, ‘M’, ‘=’, ‘S’, ‘G’, ‘]’, ‘+’, ‘[’, ‘C’, ‘A’, ‘S’, ‘E’, ‘=’, ‘A’, ‘D’, ‘E’, ‘]’)
ES	WR	(‘VMP’)
	CHAR	(‘V’, ‘M’, ‘P’)

with all the morphological tags in the analysis. This embedding configuration is called WITH ROOT (WR). This is the simplest embedding style that can be considered given the morphological analysis of words in any language. This is because most of the time the morphological tags output by morphological disambiguation tools can be treated as sequences.

For Turkish, the morphological analysis of the word ‘evlerinde’, which is ‘ev+Noun+A3pl+P3sg+Loc’, is transformed into a list by splitting from the ‘+’ symbols. For Czech, the output of the morphological disambiguator (Votrubec 2006) are the root lemma and a string of fifteen characters for tags. In the WR configuration, this is transformed into a fixed-length list by splitting with the ‘+’ symbol. For instance, the analysis ‘prezident+NNMS1-----A----’ is converted into ‘prezident’, ‘NNMS1-----A----’. A similar transformation is applied to Hungarian. The analysis of the word ‘Magyar’, ‘Magyar+PROPN+Case=Nom+Number=Sing’, is converted into the list ‘Magyar’, ‘PROPN’, ‘Case=Nom’, ‘Number=Sing’. The disambiguator that is used for Finnish outputs the tags separately so it is just transformed

into a list. The disambiguated morphological analysis of the word 'Tampereella' is 'tampere+[POS=NOUN]+[PROPER=PROPER]+[NUM=SG]+[CASE=ADE]'. Then, for the embedding configurationWR, it is transformed into 'tampere', '[POS=NOUN]', '[PROPER=PROPER]', '[NUM=SG]', '[CASE=ADE]'. One exception to this is Spanish in our choice of languages. For Spanish, a single tag is used to represent both the POS and the morphological properties that might be attached to the word. For instance, the Spanish word 'visitada' is only labeled with 'VMP', indicating the past participle form. So, the result is just a single item list for Spanish.

The second embedding configuration follows from the WITH ROOT (WR) scheme. The root is omitted from the list and the resulting list is the embedding configuration WITHOUT ROOT (WOR). Obviously, this embedding configuration does not make sense for languages that do not carry the root lemma in their morphological analysis representations, such as Spanish in our case.

The next configuration is specific to the Turkish language. The morphological notation of Turkish includes a 'DB' (derivational boundary) tag. This tag denotes a change in the POS during the derivational process. The parts that are separated by DB tags are called inflectional groups (Oflazer 2003). To examine the significance of these boundaries on the performance of the task, the tags between the root and the last derivational boundary are removed. The intuition is that the information given by the features before the last derivational boundary may not be relevant at all or may be relatively less relevant. This is because the last derivational part yields the derived word whose lexical and syntactic properties may be slightly different from the intermediate parts. This configuration is named as WITH ROOT AND AFTER LAST DB (WR_ADB). Table 1 shows an example for the word 'İstanbul'daydı' (he/she was in İstanbul), whose analysis was given in Section 3.1.1.

Finally, the morphological analysis of a word is simply treated as a string, which is transformed to a list containing each of its characters. This embedding configuration is referred to as CHAR.

5 Experiments

To test the validity and performance of our proposed method, two main set of experiments are conducted: (i) experiments that compare the proposed approach with the state-of-the-art models, and (ii) experiments that aim to demonstrate the differing performance characteristics of different model configurations. In this section, before giving the results of the experiments, the training method and the datasets used in the experiments are explained.

5.1 Training

The parameters to be learned by the training algorithm are the parameters of the Bi-LSTM described in Section 4 (Figure 2), the parameters of the Bi-LSTMs for the *character* and *morphological* embeddings (Figure 3) and the word embeddings for each unique word. After experiments with several different choices for the number of dimensions for these parameters, a choice of 100 for word embeddings, 200 for

character embeddings and 200 for morphological embeddings was observed to give the best results. However, it was not possible to use these dimension sizes in the experiments with all the languages and configurations due to time complexity.

In the first set of experiments that aim to compare the model configurations, the cell dimension of the sentence-level LSTM, word, character and morphological embedding dimensions and character and morphological LSTM cell dimensions were set as ten. For word embeddings, embeddings of size ten were trained with an algorithm that takes sub-word information into account (Bojanowski *et al.* 2017) using the corresponding language version of Wikipedia and were used as pretrained word embeddings. Higher dimension sizes that yield the best performances as stated above were used in the second set of experiments that compare the proposed model with the previous work.

Model training was done by calculating the gradients using the back propagation algorithm and updating with the stochastic gradient descent algorithm with a learning rate of 0.01. Gradient clipping was employed to handle gradients diverging from zero. Additionally, dropout was used on the inputs with probability 0.5. Each language was trained for 50 epochs.

5.2 Datasets

Five languages were selected to evaluate the proposed method on a set of MRLs: Turkish, Czech, Hungarian, Finnish and Spanish. In this section, the specifics of each dataset like the morphological analysis format, the pretrained word embeddings we used and the origin of the data is given.

Turkish: Our model is trained and evaluated using a corpus which was widely used in previous works on Turkish NER (Tür *et al.* 2003). The training part of the corpus contains 14,481 person names, 9,411 location names and 9,037 organization names while the test part contains 1,594 person names, 1,091 location names and 863 organization names. In addition to the named entity tags and the corresponding surface forms, the corpus also contains a single disambiguated morphological analysis for each input word.

Word embeddings[4] of Turkish words as vectors of length 100 were obtained using the skipgram algorithm (Mikolov *et al.* 2013) on a corpus of 951 million words, 2,045,040 of which are unique (Yildiz *et al.* 2016). This corpus consists of Turkish texts extracted from several national newspapers, news sites and book transcripts. The fastText algorithm was employed to obtain word embeddings of size ten using the same corpus (Bojanowski *et al.* 2017).

Czech: CNEC 2.0 corpus was used to test the performance of our model on the Czech language (Ševčíková *et al.* 2007; Konkol and Konopik 2013). Seven different named entity types are labeled in this corpus. The number of labels for each of these entity types for training, validation and test portions of the dataset is given in Table 2. For each word, the morphological analysis provided in the dataset was

[4] These word embeddings are available at `https://github.com/onurgu/linguistic-features-in-turkish-word-representations/releases`.

Table 2. *The number of labels for each entity type in Czech and Finnish datasets*

Czech							
	Person	Geographical	Institution	Media	Address	Time	Artificial
Training	3,757	3,117	2,705	314	402	2,431	2,459
Validation	509	431	340	53	77	280	325
Test	480	378	324	48	55	368	382

Finnish									
	Person	Location	Org.	Misc.	Date	Event	Product	Time	Title
Training	2,229	2,040	9,098	907	956	93	4,462	4,958	631
Test	409	505	1,910	182	238	17	1,134	1,066	129

used. The fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 for Czech using the Czech version of Wikipedia (Bojanowski *et al.* 2017).

Hungarian: The Named Entity Corpus for Hungarian corpus that contains around 14,400 phrases tagged with entity labels were used (Szarvas *et al.* 2006). The corpus is labeled with the standard named entity tags. The training part contains 795 person names, 1,056 location names, 8,458 organization names and 1,327 miscellaneous names. The corpus originally contained only training and test parts, so validation and test sets were created by randomly selecting from the test part. The test set contains 100 person names, 125 location names, 1,055 organization names and 160 miscellaneous names. In the validation set, there are 87 person names, 113 location names, 1,020 organization names and 174 miscellaneous names. A statistical morphological analysis tool for Hungarian was used to process each word (Zsibrita *et al.* 2013) and its output was used as the input for morphological embeddings. For word embeddings, the fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 for Hungarian using the Hungarian version of Wikipedia (Bojanowski *et al.* 2017).

Finnish: A labeled corpus[5] that was compiled from news articles in an online Finnish technology news site was used. The articles were published between 2014 and 2015. Extracting the morphological tags was done by a Finnish morphological analysis tool called Omorfi. Morphological disambiguation was done by FinnPos while creating the training and test sets (Silfverberg *et al.* 2016). This corpus is labeled with five more named entity tags in addition to the standard set: 'DATE' for depicting date references, 'EVENT' for marking events, 'PRO' for marking products, 'TIM' for marking time expressions and 'TIT' for titles. The number of labels for each entity type is shown in Table 2. The fastText algorithm was used to obtain

[5] `https://github.com/mpsilfve/finer-data`

pretrained word embeddings of size 10 and 100 for Finnish using the Finnish version of Wikipedia (Bojanowski *et al.* 2017).

Spanish: CoNLL 2002 Shared Task[6] publishes a corpus tagged with NER and POS labels, which has clearly defined training, development and testing portions of the dataset. This dataset is widely used in NER related research for benchmarking. The POS tags were treated as the morphological analysis of the word as the POS tag contains the morphological information associated with the word if there is any. This corpus contains 6,278 person names, 6,981 location names, 10,490 organization names and 2,957 names of miscellaneous types. The fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 using the Spanish version of Wikipedia[7] (Bojanowski *et al.* 2017).

5.3 Results

This section presents the results of experiments performed to measure the impact of using morphological information for the NER task along with character-based embeddings. The experiments are conducted with the Turkish, Czech, Hungarian, Finnish and Spanish languages.

The experiments are performed with alternative embedding configurations, which are referred to with *Setup* followed by an integer as an identifier. The setups are as follows: (i) only pretrained word embeddings (Setup 1), (ii) only word and character embeddings (Setup 2), (iii) only word and a choice of one of the morphological embedding configurations (Setups 3–6) and (iv) word, character and a choice of one of the morphological embeddings (Setups 7–10).

Table 3 summarizes these results. A comparison of the basic model (Setup 1) with those that use morphological information (Setups 3–6) shows a performance increase when morphological information is used (`ME(CHAR)` and `ME(WOR)`).

In the case of Setup 4, an improvement is observed only for Turkish. Also, again for Turkish, using only the tags after the last derivational boundary (`ME(WR_ADB)`) is one of the most successful morphological configurations.

The performances of `ME(CHAR)` and `ME(WOR)` are comparable with the latter being slightly lower (except for Czech). The difference in the performance between Setups 5 and 6 may stem from the errors present in the morphological analyses. These errors are mostly due to unknown or misspelled words. In such cases, the analysis in the corpus usually defaults to the same nominal case. The higher performance of `ME(CHAR)` may be attributed to the ability to handle possibly faulty roots as morphological embedding captures more useful information in comparison to `ME(WOR)`.

Another advantage of `ME(CHAR)` embeddings is its ability to capture the relationship between roots with the same prefix. For example, in the Finnish corpus, the frequencies of words with the same prefix often differ significantly based on their roots, such as in the cases of 'allekirjoittaa' (sign) *versus* 'allekirjoittaja' (signatory) and 'Tampere' (a city in Southern Finland) *versus* 'Tamperelainen' (of Tampere),

[6] `http://www.lsi.upc.es/ nlp/tools/nerc/nerc.html`
[7] `http://www.wikipedia.org`

Table 3. *The performance of the model using various embedding configurations for five languages (boldface values indicate the best values among models for the corresponding language)*

Setup	Setups		F1-Measure				
	CE	ME	TR	CS	HU	FI	ES
1	–	–	82.25	67.56	94.02	70.56	80.38
2	CE	–	86.70	72.35	95.10	79.36	81.00
3	–	ME(WR_ADB)	87.99	N/A	N/A	N/A	N/A
4	–	ME(WR)	87.78	66.62	93.98	67.30	N/A
5	–	ME(CHAR)	88.12	72.66	95.11	75.89	82.19
6	–	ME(WOR)	87.78	67.85	95.14	75.34	81.44
7	CE	ME(WR_ADB)	87.69	N/A	N/A	N/A	N/A
8	CE	ME(WR)	87.09	69.10	92.67	72.17	N/A
9	CE	ME(CHAR)	**91.04**	**73.61**	**95.60**	**81.37**	**82.94**
10	CE	ME(WOR)	89.85	67.19	95.50	80.27	82.68

where the occurrence of the former is much greater than the latter. The ME(CHAR) scheme also benefits from the common parts in related morphological tags. For instance, in Turkish, the tags 'A3sg' and 'A3pl' denote third person singular and third person plural, respectively, where the leading two characters 'A3' indicate third person agreement. The model can capture this information when the tags are represented in terms of characters. Therefore, ME(CHAR) is a better representation than either ME(WR) or ME(WOR).

Using only character embeddings in addition to the word embeddings (Setup 2) also improves the NER performance. Combining character embeddings with the ME(CHAR) and ME(WOR) models (Setups 9 and 10) outperforms all other setups (except Czech). For all the languages, the best performance is achieved with Setup 9 where all types of embeddings are employed[8]. Although adding CE to Setup 1 (word embeddings) causes a large improvement, adding CE to Setup 5 (word embeddings and character embeddings of morphological part) provides a relatively small increase in performance (Setup 9). This may be the result of CE and ME(CHAR) both taking part in representing morphological information of words.

The results of these experiments show that an increase in NER performance is observed for all languages when either of CE or ME is included in the word representation. The best performance is achieved when both CE and ME are included in the word representation for all languages.

Table 4 shows a comparison of the proposed approach with the state-of-the-art results reported in literature. The CE+ME(CHAR) configuration (Setup 9) is used for comparison, since it yielded the best results. The models for each language were trained with higher number of parameters. The values for cell dimension of

[8] The lower values for Czech are due to the corpus that was used. This is also apparent in Table 4 where we compare our best results with the literature, i.e., the performance on Czech dataset of other work is also relatively lower compared to the performance on the Turkish dataset.

Table 4. *Comparison of results with state-of-the-art NER results for each language (boldface values indicate the best values among models for the corresponding language)*

Work	F1-Measure				
	TR	CS	HU	FI	ES
(Kuru *et al.* 2016)	91.30	72.19	-	-	-
(Demir and Özgür 2014)	91.85	75.61	-	-	-
(Şeker and Eryiğit 2012)	91.94	-	-	-	-
(Lample *et al.* 2016)	-	-	-	-	85.75
(unpublished, uses Stanford NER)[10]	-	-	-	82.42	-
(Varga and Simon 2007)	-	-	94.77	-	-
(Straková *et al.* 2016)	-	80.79	-	-	-
This work	**92.93**	**81.05**	**96.11**	**84.34**	**86.95**

the sentence-level LSTM, character and morphological LSTM cell dimensions and character and morphological embedding dimensions were all set to 200. The word embeddings was set to 100. Other hyper-parameters and training related settings were unaltered. This work is the first one to report test results for Czech, Turkish, Hungarian, Finnish and Spanish. As such, comparisons are made using different studies for each language. For Turkish, a comparison with three different results are presented. The performance of Şeker and Eryiğit (2012) shown in the table was obtained using gazetteers. When such resources are not used, the performance drops to 89.55 per cent. Kuru *et al.* (2016) did not employ any external data. Demir and Özgür (2014) relied on hand-crafted features, however exploit externally trained word embeddings. For Finnish, the creators of the dataset are in the progress[9] of publishing a NER model. For Spanish, the result from Lample *et al.* (2016) is reported. A noteworthy observation is the increase in NER performance for Spanish even though its morphological characteristic differs from the other languages.

6 Conclusions

In this work, a state-of-the-art system for NER in MRLs was introduced. The impact of alternative combinations for embedding morphological tags that were examined revealed that augmenting word representations with morphological embeddings improves NER performance, which is further improved when combined with character-based word representations. Experiments with five languages, all MRLs except Spanish, were performed. The results obtained using this approach are the state-of-the-art for all of these languages. An ablation study to examine the impact of using morphological information revealed that the improved performance was similar across these languages.

[9] Personal communication.
[10] This is the only other result using this corpus and is reported by the creators of the corpus, for details see https://github.com/mpsilfve/finer-data/blob/master/documents/finer.tex.

Although extensive experiments with the interaction of morphological analysis based and surface form based representations were done, a correlation analysis was not conducted. A future work can give insight on the morphological representation quality of the character-based representation by inspecting the conditions of this correlation.

References

Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., Kameyama, M., Martin, D., Myers, K., and Tyson, M. 1995. SRI International FASTUS system: MUC-6 test results and analysis. In *Proceedings of the 6th Conference on Message Understanding*, Association for Computational Linguistics, pp. 237–48.

Babych, B., and Hartley, A. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT Workshop on MT and Other Language Technology Tools, Improving MT through Other Language Technology Tools: Resources and Tools for Building MT*, Association for Computational Linguistics, pp. 1–8.

Bhatia, P., Guthrie, R., and Eisenstein, J. 2016. Morphological priors for probabilistic neural word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 490–500.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**: 135–46.

Borthwick, A. E. 1999. *A Maximum Entropy Approach to Named Entity Recognition*, Ph.D. thesis. New York, NY, USA: New York University.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing. In *Proceedings of the 25th International Conference on Machine Learning (ICML-08)*, ACM, pp. 160–7.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**: 2493–537.

Cotterell R., and Schütze, H. 2018. Joint Semantic Synthesis and Morphological Analysis of the Derived Word. *Transactions of the Association for Computational Linguistics* **6**: 33–48.

Çöltekin, Ç. 2014. A set of open source tools for Turkish natural language processing. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pp. 1079–86.

Demir, H., and Özgür, A. 2014. Improving named entity recognition for morphologically rich languages using word embeddings. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, IEEE, pp. 117–22.

Dietterich, T. G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* **10**: 1895–923.

Erjavec, T. 2004. MULTEXT-East version 3: multilingual morphosyntactic specifications, lexicons and corpora. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-04)*, ELRA, pp. 1535–1538.

Erjavec, T. 2010. MULTEXT-East version 4: multilingual morphosyntactic specifications, lexicons and corpora. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Farkas, R., Szeredi, D., Varga, D., and Vincze, V. 2010. MSD-KR harmonizacio Szeged Treebank 2.5-ben [Harmonizing MSD and KR codes in the Szeged Treebank 2.5]. In *Proceedings of the VII Magyar Szamıtogepes Nyelveszeti Konferencia*, pp. 349–53.

Finkel, J. R., Grenager, T., and Manning, C. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 363–70.

Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM networks and other neural network architectures. *Neural Networks* **18**: 602–10.

Grishman, R., and Sundheim, B. 1996. Message understanding conference-6: a brief history. In *Proceedings of the 16th Conference on Association for Computational Linguistics*, pp. 466–71.

Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., and Su, Z. 2009. Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 281–9.

Guo, J., Xu, G., Cheng, X., and Li, H. 2009. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp. 267–74.

Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., Ševčíková-Razímová, M., and Urešová, Z. 2006. *Prague Dependency Treebank 2.0*. Philadelphia, PA, USA: Linguistic Data Consortium.

Hajič, J., Hajičová, E., Mikulová, M., and Mírovský, J. 2017. Prague dependency treebank. In N. Ide and J. Pustejovsky (eds.), *Handbook of Linguistic Annotation*, pp. 555–94. Netherlands: Springer.

Hana, J., Zeman, D., Hajic, J., Hanová, H., Hladká, B., and Jerábek, E. 2005. Manual for morphological annotation. ÚFAL Technical Report, Revision for the Prague Dependency Treebank 2.0 (No. 2005/27).

Harris, Z. S. 1954. Distributional structure. *Word* **10:** 146–62.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* **9**: 1735–80.

Huang, Z., Xu, W., and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991.

Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. 1998. University of Sheffield: description of the LaSIE-II system as used for MUC-7. In *Proceedings of the 7th Message Understanding Conferences (MUC-7)*, ACL.

Jiang, J., and Zhai, C. X. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 264–71.

Konkol, M., and Konopik, M. 2013. CRF-based Czech named entity recognizer and consolidation of Czech NER research. In I. Habernal and V. Matoušek (eds.), *Text, Speech and Dialogue*, pp. 153–60. Lecture Notes in Computer Science, vol. 8082. Berlin, Heidelberg: Springer.

Koskenniemi, K. 1983. Two-level morphology: a general computational model for word form recognition and production. Publication no. 11, Department of General Linguistics, University of Helsinki, Finland.

Koskenniemi, K. 1984. A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting on Association for Computational Linguistics*, pp. 178–81.

Kripke, S. 1982. *Naming and Necessity*. Boston: Harvard University Press.

Kuru, O., Can, O. A., and Yuret, D. 2016. CharNER: character-level named entity recognition. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING-2016)*, pp. 911–21.

Lafferty, J., McCallum, A., and Pereira, F. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pp. 282–9.

Lankinen, M., Heikinheimo, H., Takala, P., Raiko, T., and Karhunen, J. 2016. A character-word compositional neural language model for Finnish. CoRR abs/1612.03266.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. 2016. Neural architectures for named entity recognition. In *Proceedings of the Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2016), pp. 260–70.

Lee, J., Kim, G., Yoo, J., Jung, C., Kim, M., and Yoon, S. 2017. Training IBM Watson using automatically generated question-answer pairs, CoRR, abs/1611.03932.

Liu, Y., and Ren, F. 2011. Japanese named entity recognition for question answering system. In *Proceedings of the IEEE International Conference on Cloud Computing and Intelligence Systems*, IEEE, pp. 402–6.

Lee, C., Hwang, Y., and Jang, M. 2007. Fine-grained named entity recognition and relation extraction for question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2007)*, ACM, pp. 799–800.

Luong, T., Socher, R., and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL)*, pp. 104–13.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1064–74.

McCallum, A., and Li, W. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL*, Association for Computational Linguistics, pp. 188–91.

Mikolov, T., Karafiát, M., Burget, L., Cernocky, J., and Khudanpur, S. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, p. 3.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3111–9.

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1105–16.

Oflazer, K. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing* **9**: 137–48.

Oflazer, K. 2003. Dependency parsing with an extended finite-state approach. *Computational Linguistics* **29**: 515–44.

Lewis, G. L. 1991. *Turkish Grammar*. Oxford: Oxford University Press.

Pennington, J., Socher, R., and Manning, C. D. 2014. GloVe: global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, pp. 1532–43.

Pirinen, T. A. 2015. Omorfi – Free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA-2015)*, pp. 313–5.

Proszeky, G., and Tihanyi, L. 1993. Humor: high-speed unification morphology and its applications for agglutinative languages. *La Tribune Des Industries de la Langue* **10**: 28–9.

Rao, D., McNamee, P., and Dredze, M. 2013. Entity linking: Finding extracted entities in a knowledge base. In T. Poibeau, H. Saggion, J. Piskorski and R. Yangarber (eds.), *Multi-Source, Multilingual Information Extraction and Summarization*, pp. 93–115. Berlin, Heidelberg: Springer.

Sak, H., Güngör, T., and Saraçlar, M. 2007. Morphological disambiguation of Turkish text with perceptron algorithm. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 107–18.

Santos, C. D., and Zadrozny, B. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-2014)*, pp. 1818–26.

Şeker, G. A., and Eryiğit, G. 2012. Initial explorations on using CRFs for Turkish named entity recognition. In *Proceedings of the International Conference on Computational Linguistics (COLING-2012)*, pp. 2459–74.

Ševčíková, M., Žabokrtský, Z., and Krůza, O. 2007. Named entities in Czech: annotating data and developing NE tagger. In *Proceedings of the International Conference on Text, Speech and Dialogue*, pp. 188–95.

Shen, Q., Clothiaux, D., Tagtow, E., Littell, P., and Dyer, C. 2016. The role of context in neural morphological disambiguation. In *Proceedings of the Conference on Computational Linguistics (COLING-2016)*, pp. 181–91.

Silfverberg, M., Ruokolainen, T., Lindén, K., and Kurimo, M. 2016. FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish. *Language Resources and Evaluation* **50**: 863–78.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2013)*, pp. 1631–42.

Straková, J., Straka, M., and Hajič, J. 2016. Neural networks for featureless named entity recognition in Czech. In *Proceedings of the 19th International Conference on Text, Speech and Dialogue (TSD-2016)*, pp. 173–181.

Szarvas, G., Farkas, R., Felföldi, L., Kocsor, A., and Csirik, J. 2006a. Highly accurate named entity corpus for Hungarian. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Szarvas, G., Farkas, R., and Kocsor, A. 2006b. A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. In *Proceedings of the International Conference on Discovery Science*, pp. 267–78.

Toutanova, K., Klein, D., Manning, C., and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the HLT-NAACL 2003*, pp. 252–9.

Tron, V., Halacsy, P., Rebrus, P., Rung, A., Simon, E., and Vajda, P. 2006. The annotation system of HunMorph. Technical Report, The Media Research Center, Budapest University of Technology and Economics.

Tür, G., Hakkani-Tür, D., and Oflazer, K. 2003. A statistical information extraction system for Turkish. *Natural Language Engineering* **9**: 181–210.

Turian, J., Ratinov, L., and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 384–94.

Underhill, R. 1976. *Turkish Grammar*. Cambridge, MA: MIT Press.

Varga, D., and Simon, E. 2007. Hungarian named entity recognition with a maximum entropy approach. *Acta Cybernetica* **18**: 293–301.

Votrubec, J. 2006. Morphological tagging based on averaged perceptron. In *Proceedings of the 15th Annual Conference of Doctoral Students (WDS-2006)*, pp. 191–5.

Voutilainen, A. 2011. FinnTreeBank: creating a research resource and service for language researchers with constraint grammar. In *Proceedings of NoDaLiDa 2011 Workshop on Constraint Grammar Applications*, pp. 41–9.

Wu, D., Lee, W. S., Ye, N., and Chieu, H. L. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2009)*, ACL, pp. 1523–32.

Xu, Y., and Liu, J. 2017. Implicitly incorporating morphological information into word embedding. CoRR abs/1701.02481.

Yang, Z., Salakhutdinov, R., and Cohen, W. 2016. Multi-task cross-lingual sequence tagging from scratch. CoRR abs/1603.06270.

Yeniterzi, R. 2011. Exploiting morphology in Turkish named entity recognition system. In *Proceedings of the Association for Computational Linguistics Student Session (ACL-2011)*, pp. 105–10.

Yildiz, E., Tirkaz, C., Sahin, H. B., Eren, M. T., and Sonmez, O. 2016. A morphology-aware network for morphological disambiguation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, AAAI Press, pp. 2863–9.

Zsibrita, J., Vincze, V., and Farkas, R. 2013. magyarlanc: A toolkit for morphological and dependency parsing of hungarian. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP-2013)*, pp. 763–71.