Expert Systems With Applications 182 (2021) 115049

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An evaluation of recent neural sequence tagging models in Turkish named entity recognition

Gizem Aras [a,*], Didem Makaroğlu [a,b], Seniz Demir [c], Altan Cakir [d,e]

[a] Department of Big Data and Analytics, Demiroren Teknoloji A.S., Istanbul, Turkey
[b] Informatics Institute, Istanbul Technical University, Istanbul, Turkey
[c] Department of Computer Engineering, MEF University, Istanbul, Turkey
[d] Physics Engineering, Faculty of Science and Letters, Istanbul Technical University, Istanbul, Turkey
[e] Istanbul Technical University Artificial Intelligence, Data Science Research and Application Center, Istanbul, Turkey

ARTICLE INFO

Keywords:
Named entity recognition
Turkish
Transfer learning
CRF
Digital media industry

ABSTRACT

Named entity recognition (NER) is an extensively studied task that extracts and classifies named entities in a text. NER is crucial not only in downstream language processing applications such as relation extraction and question answering but also in large scale big data operations such as real-time analysis of online digital media content. Recent research efforts on Turkish, a less studied language with morphologically rich nature, have demonstrated the effectiveness of neural architectures on well-formed texts and yielded state-of-the art results by formulating the task as a sequence tagging problem. In this work, we empirically investigate the use of recent neural architectures (Bidirectional long short-term memory (BiLSTM) and Transformer-based networks) proposed for Turkish NER tagging in the same setting. Our results demonstrate that transformer-based networks which can model long-range context overcome the limitations of BiLSTM networks where different input features at the character, subword, and word levels are utilized. We also propose a transformer-based network with a conditional random field (CRF) layer that leads to the state-of-the-art result (95.95% f-measure) on a common dataset. Our study contributes to the literature that quantifies the impact of transfer learning on processing morphologically rich languages.

1. Introduction

Named entity recognition (NER) aims to recognize named entities in a given text by determining their boundaries and classifying them into predefined categories (e.g., person, location, and temporal expression). NER is a crucial step in various natural language processing applications such as event extraction (Chen, Xu, Liu, Zeng, & Zhao, 2015) and question answering (Mollá, van Zaanen, & Smith, 2006) as well as in big data analytics (Saju & Shaja, 2017). Early studies have addressed the recognition of named entities as a sequence labeling problem and extensive research efforts have been devoted to developing solutions using machine learning techniques (Lin, Peng, & Liu, 2006; Ekbal et al., 2008), hidden markov models (Zhou & Su, 2002), and conditional random fields (Yao, Sun, Li, Wang, & Wang, 2009; Zirikly & Diab, 2015). Recently, neural models have been introduced to named entity task in well-formed and noisy texts (Al-Nabki, Fidalgo, Alegre, & Fernández-Robles, 2020). In spite of recent advances, NER remains to be a challenging problem due to several reasons such as the recognition of overlapping or nested entities, infrequent entities in user generated noisy texts, and semantically ambiguous entities in different contexts.

In the current era, the amount of online content has exploded which makes it exhaustive to search from a vast distributed source of information. Search tools or expert systems might effectively alleviate the problem of accessing available content on the web. However, continuous alteration of natural languages due to heavy social media usage, social-cultural factors in society, daily events (e.g., political changes and major sport events) has reflections in written texts and leads to constant evolution of words, expressions and importantly named entities. Correctly identified named entities from unstructured or semi-structured content form a basis for the development of more effective and intelligent information management, text mining, and relation extraction systems (Marrero, Urbano, Sanchez-Cuadrado, Morato, &

* Corresponding author.
E-mail addresses: gizem.aras@demirorenteknoloji.com (G. Aras), makaroglu17@itu.edu.tr (D. Makaroğlu), demirse@mef.edu.tr (S. Demir), altan.cakir@itu.edu.tr (A. Cakir).

https://doi.org/10.1016/j.eswa.2021.115049
Received 7 May 2020; Received in revised form 2 April 2021; Accepted 14 April 2021
Available online 3 May 2021

Gomez-Berbis, 2013). For instance, mining daily news content by digital media applications for extracting information about a person or a location necessitates querying an astonishing amount of news articles which can be facilitated by automatic detection of named entities in written texts. Paving the road for interpretable and reusable information through semantically annotated online content can also be listed as a particular benefit of extracting named entities and their relations from raw texts.

NER is a well-studied task for several languages including Turkish and recent successes in neural architectures have greatly advanced achieved performances on recognizing Turkish named entities (Güneş & Tantuğ, 2018; Güngör, Güngör, & Üsküdarli, 2019). In these studies, Bidirectional Long Short-Term Memory networks with different word representations were widely used and evaluated on a common dataset consisting of person, location, and organization names (Tür, Hakkani-Tür, & Oflazer, 2003). A conditional random field (CRF) was shown to positively contribute to these networks that minimize the need for feature engineering. There is a recent interest in applying deep bidirectional transformers (BERTurk, 2020) and transfer learning (Akkaya & Can, 2020) to Turkish entity tagging. In this work, we present a comprehensive evaluation of two notable neural architectures, namely BiLSTM networks and Transformer-based networks and compare their performances in the same experimental setting. In BiLSTM models, we explore different combinations of four kinds of embeddings as input (i.e., character, morphological, subword, and word embeddings) and experiment with different pretrained embeddings as initializations of word embeddings. In transformer-based models, we benefit from three different transformer based language models, namely multilingual cased BERT (mBERT), Turkish BERT (BERTurk), and XLM-RoBERTa (XLMR), and study the effectiveness of both linear and CRF layers at the top of the network. As our second contribution, we propose a transformer-based neural architecture accompanied with a CRF as the top layer (an extension of the BERTurk model) which sets the new state-of-the-art f-measure of 95.95%. Our study not only extends the current Turkish NER literature but also validates the usability of transfer learning on processing a morphologically rich language.

The rest of this article is organized as follows. Section 2 discusses related research on named entity recognition with a particular focus on Turkish NER studies. Section 3 describes neural architectures utilized in this work. Section 4 presents our dataset and parameter initializations used for building neural architectures. Section 5 discusses conducted experiments and the results that we obtained. Finally, Section 6 concludes the article and presents our future work.

2. Literature review

2.1. Neural models for named entity recognition

Earlier traditional named entity recognition systems have relied heavily on feature engineering and employed hand-crafted language dependent features, large gazetteers, and tagged datasets (Collobert et al., 2011). A significant branch of research has utilized a range of statistical approaches to address the problem such as maximum entropy classifiers (Chieu & Ng, 2003), decision trees (Paliouras, Karkaletsis, Petasis, & Spyropoulos, 2000), and conditional random fields (Finkel & Manning, 2009). However, in recent years, the focus of NER research has shifted to neural models in parallel with observed improvements on multiple language processing benchmarks such as question–answering and language generation. Neural NER models have been guided by distributional approaches where the meaning of a word is carried in its surroundings via vector representations (Harris, 1954; Firth, 1957; Mikolov, Chen, Corrado, & Dean, 2013). Initial attempts considered words as separate tokens and represented each token using a fixed-length vector (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Pennington, Socher, & Manning, 2014). Some other studies explored different ways of representing words such as concatenating embeddings of characters (dos Santos & Guimarães, 2015), morphemes (Luong, Socher, & Manning, 2013), or other word subparts to fixed-length word embeddings. In recent NER studies, the problem was formulated as a sequence labeling task and different Seq2Seq models were shown to achieve state-of-the art results where final embeddings of words are encoded by gated recurrent units (GRUs) or long short-term memory units (LSTMs) (Ma & Hovy, 2016; Chen & Moschitti, 2018). Using conditional random fields on top of neural networks were proved to work equally well (Collobert et al., 2011; Huang, Xu, & Yu, 2015; Chiu & Nichols, 2016) or better than previous methods. Moreover, BiLSTM-CRF models with character and word embeddings were shown to be effective for multiple languages including Chinese (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016; Zhang & Yang, 2019) and arguably considered as a base model for tagging (Jurafsky & Martin, 2018). Unfortunately, a word, no matter in which context it appears, is represented with the same final embedding in these models. A recent study has utilized contextual string embeddings (Akbik, Blythe, & Vollgraf, 2018), where final word embeddings are contextualized according to the entire sentence. In that study, all characters in the sentence up to the last character of a word were processed via a forward LSTM and all characters from the end to the beginning of the sentence were processed via a backward LSTM. The obtained hidden states were then concatenated to produce final embedding of the word in focus. This kind of word embeddings was proved to improve not only NER tagging but also other sequence labeling tasks such as part-of-speech labeling and phrase chunking. Achieved performance scores (f1 scores) of some these English NER studies are given in Table 1.

Transformer-based approaches outperformed the state-of-the-art on several NLP tasks and achieved performance improvements that might be attributed to the use of attention-mechanism (Vaswani et al., 2017). Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018) is a bi-directional transformer that learns contextualized input representations. BERT is different from earlier work in four aspects. First, it uses transformers (Vaswani et al., 2017) instead of LSTMs to encode inputs. Second, its training objective is masked language modeling and hence instead of predicting the next word, BERT predicts a randomly masked word from a given sentence. Third, BERT uses subword tokens instead of word tokens, thus some infrequent words get eliminated and their common sub-parts are utilized[1]. Finally, the pre-trained language model can be fine-tuned for a specific language task at hand by adding one last layer on top of the utilized neural architecture. Thereafter, multilingual BERT (mBERT) was released to support many languages in a single model. However, some research studies showed that BERT trained for a single language outperforms mBERT in several tasks such as dependency parsing and natural language inference (Martin et al., 2019). Robustly optimized

Table 1
English NER studies.

Study	Approach	Embedding	F1 Score
Ma and Hovy (2016)	CNNChar-BiLSTM-CRF	-	80.76
Collobert et al. (2011)	Tanh-CRF	-	81.47
Huang et al. (2015)	BiLSTM-CRF	-	84.26
Huang et al. (2015)	BiLSTM-CRF	Senna	84.74
Ma and Hovy (2016)	CNNChar-BiLSTM-CRF	Skip-Gram	84.91
Collobert et al. (2011)	Tanh-CRF	Senna	88.67
Huang et al. (2015)	BiLSTM-CRF	Senna	88.83
Ma and Hovy (2016)	CNNChar-BiLSTM-CRF	Senna	90.28
Lample et al. (2016)	LSTMChar-BiLSTM-CRF	Skip-Gram	90.96
Ma and Hovy (2016)	CNNChar-BiLSTM-CRF	Glove	92.21
Akbik et al. (2018)	Flair-Char-BiLSTM-CRF	Glove	93.09

[1] Word sub-parts were shown to reduce data sparsity problem in morphologically rich languages such as German (Kudo & Richardson, 2018), and Turkish (Akkaya & Can, 2020)

BERT pretraining (Liu et al., 2019) demonstrated that longer training with careful hyperparameter selection could achieve better results as compared to earlier studies. Another transformer XLM-RoBERTa (XLMR) combined robustly optimized BERT pretraining approach with cross lingual language pretraining (Lample et al., 2019), while using a larger dataset, and outperformed mBERT in several tasks. In other studies, transformer based architectures were both explored with and without the addition of a CRF layer. For instance, named entity recognition in Slavic languages (Arkhipov, Trofimova, Kuratov, & Sorokin, 2019) and in Portuguese were confirmed to be improved once a trained BERT model is accompanied with a CRF layer (Souza, Nogueira, & Lotufo, 2019).

2.2. Turkish named entity recognition

Turkish is an agglutinative language with rather complex morphotactics where a lot of information is encoded (such as syntactic roles and relations of words) in morphology. Several Turkish words can be derived by appending multiple suffixes (i.e., inflectional and derivational) to a nominal or verbal root, as often seen in other morphologically rich languages such as Finnish, Hungarian, and Czech. The morphological structure of a Turkish word can be represented as a sequence of inflectional morphemes (IG) separated by derivation boundaries(DB). Each IG sequence has its own part of speech (POS) and inflectional features, and the beginning of a new sequence is marked by a derivation boundary where a change in part of speech occurs. A word might have multiple such representations due to morphological ambiguity. For instance, the following is one possible representation of the word "haberleşmeliyiz" (*we should communicate*) where the first IG represents that the root is a verb and it is transformed into a noun with the addition of the 2nd infinitive suffix "-me":

haberles + Verb + Pos
^DB + Noun + Inf2 + A3sg + Pnon + Nom
^DB + Adj + With
^DB + Noun + Zero + A3sg + Pnon + Nom
^DB + Verb + Zero + Pres + A1pl

Although surface forms are constrained by morphological rules (e.g., vowel harmony and vowel drops) (Oflazer, Gocmen, & Bozsahin, 1994), the number of derived words from a single root is still very large to be handled easily and lexical sparsity is often experienced in learning-based NLP applications. For instance, in a Turkish dataset of 10 million words, the vocabulary size is measured as 474,957 whereas that number is lowered to 97,734 unique words in an English dataset of the same size (Hakkani-Tür, 2000). However, the vocabulary size is observed to degrade to 94,235 unique words once the root forms of Turkish words are considered over the same dataset. As a common practice to handle data sparsity, Turkish NLP studies often utilize disambiguated morphological representations of words rather than their surface forms.

Named Entity Recognition in Turkish has been studied for many years (Kucuk, Arici, & Kucuk, 2017). The first statistical Turkish NER study (Tür et al., 2003) trained an HMM model to tag person, location, and organization names that appear in well-written texts by leveraging morphological, lexical and contextual information of words. In another study (Kucuk & Yazici, 2010), a rule-based approach was explored where knowledge resources such as dictionaries of person and location names, and pattern extraction rules for temporal and numeric expressions are heavily utilized. The system was then enriched with an ability to learn knowledge resources from annotated data. A CRF based NER system (Yeniterzi, 2011) highlighted the impact of morphology on tagging process and benefited from roots and morphological features of words as separate tokens instead of words. An automated rule learning system (Tatar & Cicekli, 2011), a CRF based system relying on the use of gazeteer and hand crafted morphology dependent features (Sker & Eryigit, 2012), and a classification system where six different models are trained with both discrete and continuous features of words (Ertopcu et al., 2017) are among recent Turkish NER studies. Although we use the same dataset for training and testing purposes (Tür et al., 2003), our work utilizes a neural network based solution and hence significantly differs from these earlier rule-based or statistical approaches.

The first neural network based study (Demir & Özgür, 2014) used a regularized average perceptron algorithm and combined continuous vector representations of words and some language independent features (such as context, previous tags, and case features) in a semi-supervised fashion. The use of character embeddings rather than word embeddings was later explored in a stacked bidirectional LSTM network (Kuru, Can, & Yuret, 2016). For each input character, the system outputs a tag probability and a Viterbi decoder converts character-level probabilities to word-level tag probabilities. The results demonstrated that a good tagging performance could be achieved without benefiting from an extensive list of word features and language dependent knowledge resources. The current state-of-the-art systems utilized bidirectional LSTM networks and experimented with different word representations. The first BiLSTM study (Güngör et al., 2019) concatenated word, character, and morphological embeddings as encoder inputs and used a CRF layer on top of the decoder. The tagging model was tested on three other morphologically rich languages (i.e., Czech, Hungarian, Finnish; and additionally Spanish) and the results demonstrated that word representations once augmented with morphological and character embeddings achieve the highest performance. On the other hand, the second BiLSTM study (Güneş & Tantuğ, 2018) combined word embeddings and writing style embeddings (e.g., all in uppercase letters or in sentence case letters) as input representations, and experimented with stacked layers of varying depth. In a hierarchical multi-task learning setting (Akdemir, Shibuya, & Güngör, 2020), the NER task was handled along with dependency parsing by utilizing contextual subword embeddings. A CRF layer on top of a Highway-LSTM architecture (HLSTM) was used for tagging named entities. There is only one work where deep bidirectional transformers were utilized (BERTurk, 2020), and in that study both cased and uncased BERT models were evaluated on Turkish NER task. The performances of these systems are listed in Table 2. Our work lies on the path opened by BiLSTM studies where different embeddings are learned and sequentially encoded by LSTMs. However, to the best of our knowledge, this work is the first Turkish NER study that compares language models learned by transformers with BiLSTM models in the same experimental setup. Moreover, our work explores the impact of context on task performance by exploring both context sensitive and insensitive word embeddings.

In recent years, another branch of Turkish NER studies has focused on noisy data specifically from social media. Although a limited number of approaches (Çelikkaya, Torunoğlu, & Eryiğit, 2013; Eken & Tantug, 2015; Okur, Demir, & Ozgur, 2016; Akkaya & Can, 2020) have provided different solutions to the task, they all continuously improved on the state of the art. The current state of the art with an f-score of 67.39% is still behind the observed performances on clean data.

3. System architecture

Named entity recognition is a labeling task over a text that consists of a sequence of words, and hence any approach that tags every single word with a label from a predetermined set would be a reasonable solution. In this work, we address the task as a sequence to sequence

Table 2
Turkish NER studies.

Study	Approach	F1 Score
Kuru et al. (2016)	Stacked BiLSTM	91.30
Demir and Özgür (2014)	Reg. Avg. Percp.	91.85
Güngör et al. (2019)	BiLSTM-CRF	92.93
Güneş and Tantuğ (2018)	Deep-BiLSTM	93.69
Akdemir et al. (2020)	HLSTM	93.82
BERTurk (2020)	BERT	95.40

(Seq2Seq) learning problem and build two different architectures for tagging. The first architecture utilizes a Bidirectional Long Short-Term Memory (BiLSTM) network whereas the second architecture uses a Transformer-based neural network. A CRF layer is employed on top of these architectures as an optimization layer for predicting the best label sequence. Our study has similarities with some earlier works (Güneş & Tantuğ, 2018; Güngör et al., 2019; Akkaya & Can, 2020), but the main difference comes from the utilization of a context-sensitive language model and its performance comparisons with well-studied LSTM based language models.

3.1. BiLSTM network

LSTM networks are specialized forms of recurrent neural networks that can cope with the vanishing gradient problem (Hochreiter & Schmidhuber, 1997). BiLSTM architectures utilize two separate LSTM networks where the first network processes input in the forward direction to keep a history from the beginning of the sequence whereas the second network processes all words in the sequence starting from the end of the input.

Our problem is formulated as given an input sentence $S=\{s_1,s_2,\ldots,s_n\}$ consisting of n words, obtain a sequence of labels $L=\{l_1,l_2,\ldots,l_n\}$ such that each l_m is from a set of NER tags. As shown in Fig. 1, a sequence of encoded words (x_e), i.e., embeddings, are fed to the network. In the current implementation, each word is encoded by a combination of four different embeddings:

- Word embedding: Vector representation of the word as a token (w_e)
- Subword embedding: Vector representation of the word chunk as a token (sw_e)
- Character embedding: Vector representation of the word at character-level (c_e)
- Morphological embedding: Vector representation of the word at morphological-level (m_e)

We use a context-insensitive language model to obtain the word embedding (w_e) of each token, where every word in the sequence is taken as a single token. This embedding neither captures the location of the word in the sequence nor the contextual content of the input. We obtain subword, character, and morphological embeddings of words using distinct BiLSTM networks. For instance, the network that produces character embeddings processes every character in a word as a separate token, as shown in Fig. 2-a. On the other hand, morphological BiLSTM network with a similar architecture produces embeddings to reflect morphological subunit information of each word in the sequence. Subword embeddings exploit the highest possible similarity between different words. These four kinds of embeddings are utilized in order to capture the morphologically-rich nature of Turkish and information encoded in terms of characters, morphemes, and word chunks. Separate embeddings allow us to explore different ways of concatenating them to obtain the final input embedding used by our architecture. For instance, model shown in Fig. 2-b, -b, concatenates word, character, and morphological embeddings in order to obtain the final input word embedding (i.e., $x_e = w_e \oplus c_e \oplus m_e$).

The computations performed in our architecture with LSTM cells are as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi}) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf}) \\
\mathbf{g}_t &= \tanh(\mathbf{W}_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg}) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{io}\mathbf{x}_t + +\mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho}) \\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{c}_t)
\end{aligned} \tag{1}$$

where $\mathbf{h}_t$ is the hidden state, $\mathbf{c}_t$ is the cell state, $\mathbf{x}_t$ is the input at time t, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{g}_t$, and $\mathbf{o}_t$ are the input, forget, cell, and output gates, respectively.

3.2. Transformer-based network

Transformer-based language models replace recurrent neural network cells with self attention and fully connected layers. As a result, the content of a whole sentence and the location of each word in the sentence are effectively captured to encode contextual information and long-range dependencies. Conditioning on both the left and right contexts of a word results in dissimilar encodings for the same word in different sentences. Moreover, these models enable us to benefit from shared embeddings between multiple natural languages and subword units in monolingual settings. In this architecture, we use pretrained masked language models and fine tune them on the NER task. As show in Fig. 3, the input sequence is first tokenized into subword units and then fed to the network.

3.3. CRF layer

The CRF layer is utilized as the top hidden layer in both architectures. This layer takes the concatenation of last hidden states from the underlying network. Its role is modeling the joint probability of the entire label sequence, in order to impose constraints over neighbour tokens (Lafferty, McCallum, & Pereira, 2001). A standard implementation is carried out (Zhang & Yang, 2019) and the probability of a label sequence $L = l_1, l_2, \ldots, l_n$ is calculated as follows:

$$P(L|S) = \frac{exp(\sum_i(\mathbf{W}_{Crf}^{l_i}\mathbf{h}_i + b_{Crf}^{(l_{i-1},l_i)}))}{\sum_{L'} exp(\sum_i(\mathbf{W}_{Crf}^{l_i'}\mathbf{h}_i + b_{Crf}^{(l_{i-1}',l_i')}))} \tag{2}$$

where L' represents an arbitrary label sequence, and $\mathbf{W}_{Crf}^{l_i}$ is a model parameter specific to l_i, and $b_{Crf}^{(l_{i-1},l_i)}$ is a bias specific to l_{i-1} and l_i.

For decoding, a first-order Viterbi algorithm is used to find the most probable label sequence over the input sequence, and sentence-level log-likelihood loss with L_2 regularization is used to train the model:

$$L = \sum_{i=1}^{N} log(P(y_i|s_i)) + \frac{\lambda}{2}||\Theta||^2 \tag{3}$$

where $\{(s_i, l_i)\}|_{i=1}^N$ is a set of manually labeled data, λ is the L_2 regularization parameter, and Θ is the parameter set.

4. Experimental setup

4.1. Dataset

The dataset used in this study (Tür et al., 2003) is a collection of articles from the national newspaper Milliyet, covering a period between 1 January 1997 and 12 September 1998. The dataset contains Turkish sentences tagged with BIO2 scheme in CoNNL format and morphological analyses of all sentence tokens. For instance, the sentence from the dataset "Meliha Düzağaç'in resimleri 7 Ekim'e dek Ankara TCDD Sanat Galerisi'nde sergilenecek." (*Meliha Düzağaç's paintings will be exhibited at Ankara TCDD Arts Gallery until 7th of October.*) is tagged as follows:

```
Meliha B-PERSON
Düzağaç'in I-PERSON
resimleri O
7 O
Ekim'e O
dek O
Ankara B-ORGANIZATION
TCDD I-ORGANIZATION
Sanat I-ORGANIZATION
Galerisi'nde I-ORGANIZATION
sergilenecek O
. O
```

In BIO2 scheme tagging, the first token of a named entity of a

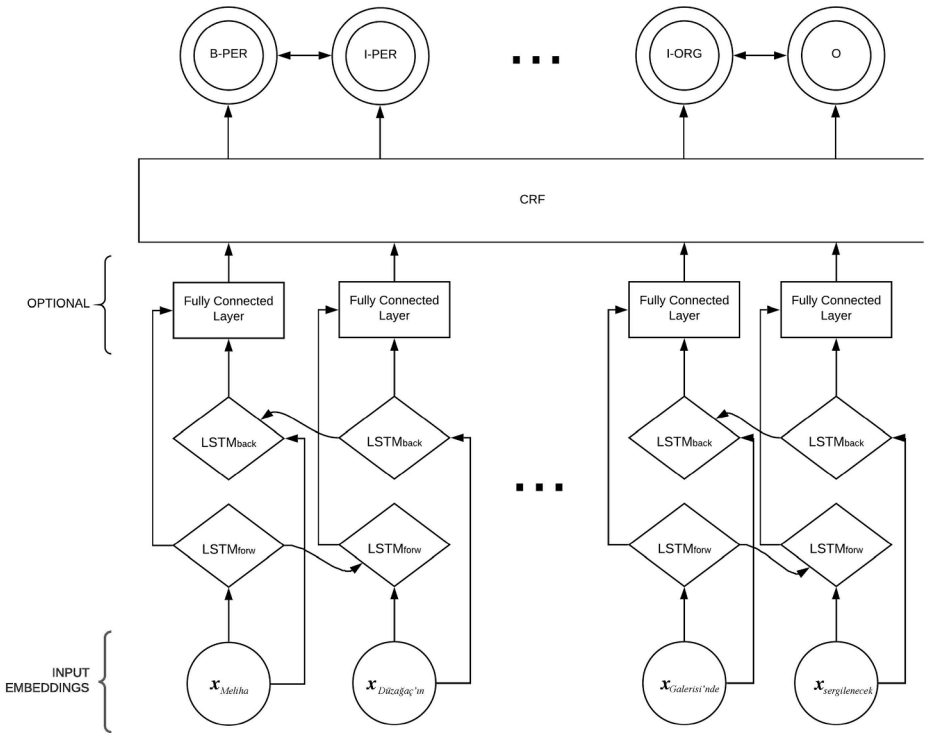

Fig. 1. The BiLSTM-CRF architecture.

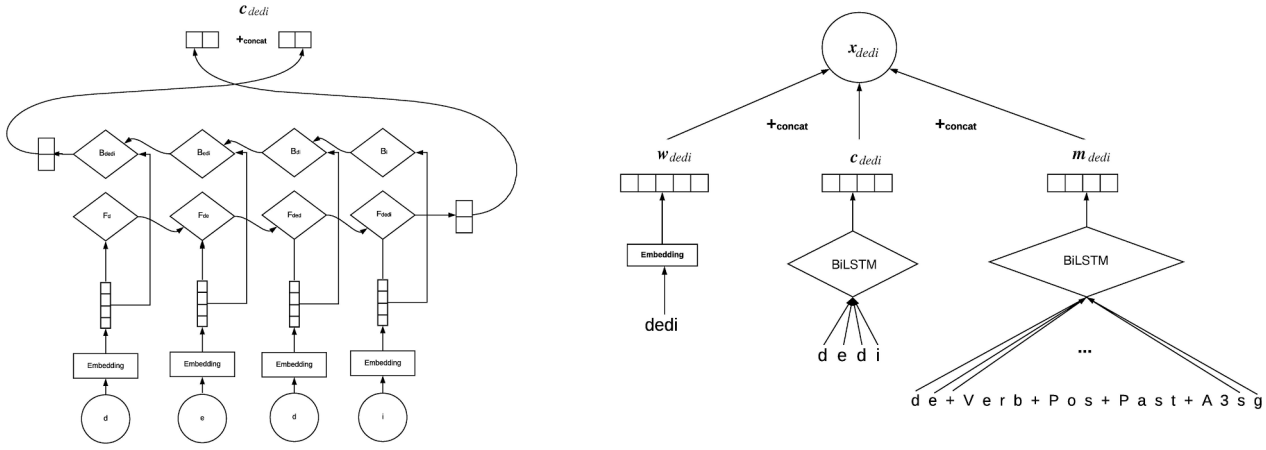

Fig. 2. (a) Character and (b) input embedding of the word "dedi" (*he/she said*).

particular type (type$_x$) is labeled with beginning type tag (B-type$_x$), and the remaining tokens of the same named entity are labeled with the inside type tag (I-type$_x$). All other tokens that do not belong to a named entity are labeled with outside type tag (O). In this work, we split the dataset into a training set of 32,171 sentences, 20% of which is reserved as validation set, and a test set of 3328 sentences.

4.2. Building BiLSTM networks

To generate input embeddings of the encoder, we first obtain vector representations of each token in our dataset. For character, morphological, and subword embeddings, vectors are randomly initialized whereas four different initializations are experimented in word embeddings (Table 3):

- *Hur:* Embeddings trained by applying word2vec skip-gram model (Mikolov et al., 2013) to articles published in the national newspaper Hurriyet from 1997 to 2019.
- *Huaw:* Skip-gram embeddings used in a previous research study (Güngör et al., 2019) where embeddings are trained with a larger dataset and window size.[2]
- *FastText:* Embeddings obtained by applying continuous bag of words model with position weights (Grave, Bojanowski, Gupta, Joulin, & Mikolov, 2018) to Common Crawl and Wikipedia dumps.[3]
- *Random:* Randomly initialized embeddings.

For each character of an input token, a random embedding is

[2] Retrieved from https://github.com/onurgu/linguistic-features-in-turkish-word-representations/releases

[3] Retrieved from https://fasttext.cc/docs/en/crawl-vectors.html#models

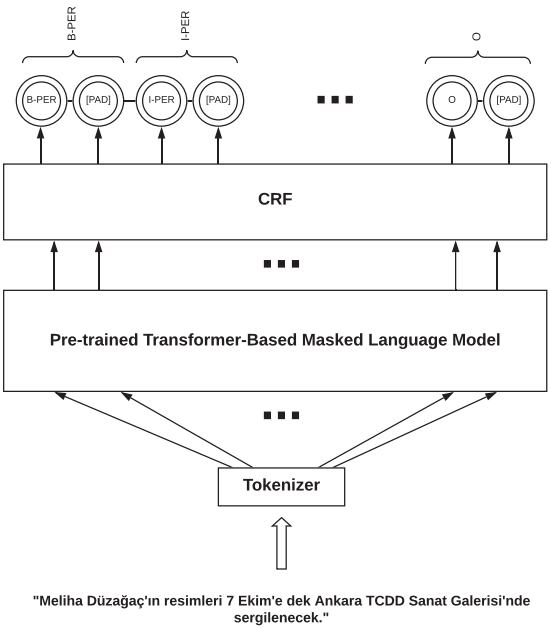

Fig. 3. The transformer-based network.

initialized. The embeddings that are formed for the token's character sequence are then fed into a bidirectional character-LSTM for encoding. BiLSTM network concatenates the forward and backward representations of the last layer as shown in Fig. 2-a. Morphological embedding of each token is formed by benefiting from its morphological analysis provided in our dataset. Character-level morphological embeddings are used, as this representation was shown to work best in a previous work (Güngör et al., 2019). Morphological embeddings have the same architecture as character embeddings, except they encode the full morphological analysis of the given word instead of the word itself (Fig. 2-b). Word chunks used in subword embeddings are obtained via a unigram SentencePiece (Kudo & Richardson, 2018) tokenizer.[4] The SentencePiece tokenizer is trained using a news archive of 14,995,202 tokens which consists of articles published in Hürriyet newspaper from 22 November 2018 to 22 November 2019. The unigram tokenizer, that we refer to as Turkish SentencePiece (TR SentPiece) tokenizer, has a vocabulary of size 50,000 tokens.

In our experiments, we use an embedding size of 300 for words and subwords whereas 200 for characters and morphological units. Using different combinations of these embeddings, we train several BiLSTM networks using stochastic gradient descent optimizer with an initial learning rate of 0.05 (some of which are listed in Table 5). In these trainings, gradient clipping of 0.5 is used and dropout is applied to concatenated embeddings with the probability of 0.5. Each models is trained for 50 epochs and 0.9 momentum is used within the optimizer. Moreover, learning rate decay is applied at the end of every epoch using the following function:

$$lr = lr_{previous} * (1/(1 + 0.05 * epoch))$$

4.3. Building transformer-based networks

To build our transformer-based networks, we utilize pretrained language models, namely multilingual cased BERT (mBERT), Turkish BERT (BERTurk)[5], and XLM-RoBERTa (XLMR). For each model, we experiment with two different settings:

- The model is followed with a linear layer and cross-entropy is used as the loss function
- The model is followed with a CRF layer and negative log is used as the loss function

In both settings, finetuning is applied to language models and sentences are tokenized by default tokenizers. However, in the first setting, subwords that do not appear in the first position of words are treated as padding tokens in loss calculations of training. In the evaluation phase, a BIO2 tag is assigned to only first subword token of a word and the remaining subwords (treated as padding in training) are labeled with the same tag. It is worth mentioning that default tokenizers provided with language models produce different subword tokens for the same sentence. For instance, outputs produced by all tokenizers used in this study for the sentence "Meliha Düzağaç'in resimleri 7 Ekim'e dek Ankara TCDD Sanat Galerisi'nde sergilenecek." are as follows:

```
Morphological Analysis:[6]
[``Meliha", ``Düzağaç", `` ' ", ``+in", ``resim",
``+ler", ``+i", ``7", ``Ekim", `` ' ", ``+e", ``dek",
``Ankara", ``TCDD", ``Sanat", ``Galeri", ``+si", `` '
", ``+n", ``+de", ``ser", ``++gi", `++len",
``+ecek", "."]

BERTurk Tokenizer:
[``Melih", ``##a", ``Düz", ``##agaç", `` ' ", ``in", ``resimleri",
  ``7", ``Ekim", `` ' ", ``e", ``dek", ``Ankara", ``TCDD", ``Sanat",
  ``Galerisi", ``'", ``nde", ``sergilen", ``##ecek", ``."]

mBERT Tokenizer:
[``Mel", ``##ih", ``##a", ``D", ``##üz", ``##a", ``##ga", ``##ç",
  `` ' ", ``in", ``res", ``##im", ``##leri", ``7", ``Ekim", `` ' ",
  ``e", ``dek", ``Ankara", ``TC", ``##D", ``##D", ``Sanat",
  ``Gale",``##risi", `` ' ", ``nde", ``ser", ``##gile", ``##nec",
  ``##ek", ``."]

XLMR Tokenizer:
[``_Meli", ``ha", ``_Düz", ``aga", ``ç", `` ' ", ``in", ``_resim",
  ``leri", ``_7", ``_Ekim", `` ' ", ``e", ``_de", ``k", ``_Ankara",
  ``_TC", ``DD", ``_Sanat", ``_Galeri", ``si", `` ' ", ``nde",
  ``_sergi", ``lenecek", ``."]

TR SentPiece Tokenizer:
[``_Melih", ``a", ``_Düz", ``agaç", `` ' ", ``in", ``_resimleri",
  ``_7", ``_Ekim", `` ' ", ``e", ``_dek", ``_Ankara", ``_TCDD",
  ``_Sanat", ``_Galerisi", `` ' ", ``nde", ``_sergilenecek", ``."]
```

In our preliminary experiments, we observe that tokens produced by multilingual BERT tokenizer do not correlate well with morphological units given in the dataset. Although this is not the case for other tokenizers, BERTurk has a small vocabulary size and XMLR is not trained solely for Turkish language. Due to these reasons, we do not report any results on the use of default tokenizers in BiLSTM networks. Moreover, in our preliminary experiments, we observe around 20%–40% mismatches between subword tokens obtained by our trained Tr SentPiece tokenizer and vocabularies used in pretrained models. Thus, we do not report results regarding the use of Tr SentPiece tokenizer in any of our transformer-based networks. HuggingFace transformers library[7] with PyTorch (Wolf et al., 2019) is used for implementations. Networks are trained by Adam optimizer with fixed weight decay, with initial learning rate of 5e −05, and gradient clipping of 1. Table 4 provides details of all language models used in transformer-based networks.

[4] Using https://github.com/google/sentencepiece
[5] https://huggingface.co/dbmdz/bert-base-turkish-cased

[6] + and ++ represent inflectional and derivational suffixes, respectively.
[7] https://github.com/huggingface/transformers

Table 3
Sets of word embeddings.

Name	Training Method	Dataset Size	Vocabulary Size	Dimension	Window Size	Negative Sampling
Hur	Skip-gram	~ 170 M	500 K	128	1	2
Huaw	Skip-gram	941 M	1.2 M	300	5	10
FastText	Continuous bag of words	-	2 M	300	5	10
Random	Randomly initialized	-	1.2 M	300	-	-

Table 5
Performance scores of BiLSTM-CRF models on our validation and test sets.

Model #	Model Description	Embedding	Valid\Test	F1	Precision	Recall	Accuracy	Trn. Time
1	Word-Char-BiLSTM-CRF	Random	Valid	85.75	84.41	87.15	97.84	11:08:25
			Test	85.20	84.10	86.33	97.76	
2	Word-Char-BiLSTM	Huaw	Valid	86.08	83.59	88.72	98.21	02:07:40
			Test	85.28	83.06	87.62	98.16	
3	Subword-Char-BiLSTM-CRF	Random	Valid	86.26	85.29	87.25	97.09	06:28:04
			Test	86.37	84.93	87.85	97.10	
4	Word-Char-BiLSTM-CRF	Hur	Valid	87.21	86.14	88.19	98.04	01:59:36
			Test	87.92	87.30	88.56	98.09	
5	Word-BiLSTM-CRF	Huaw	Valid	89.10	89.77	88.44	98.36	01:12:20
			Test	88.70	89.70	87.73	98.26	
6	Word-Char-BiLSTM-CRF	FastText	Valid	89.44	88.36	90.55	98.38	02:15:58
			Test	89.99	89.39	90.60	98.41	
7	Word-Char-Morph-BiLSTM-CRF	Huaw	Valid	91.52	90.58	92.48	98.70	05:17:29
			Test	91.65	91.38	91.92	98.71	
8	Word-Char-BiLSTM-CRF	Huaw	Valid	91.57	90.64	92.52	98.72	02:00:39
			Test	91.84	91.17	92.52	98.78	

4.4. Evaluation metrics

In this study, the evaluation scores are reported both at the token and entity levels. As a first step, the boundaries of all named entities in test sentences are determined by grouping tokens that form a single entity (i.e., a token sequence with B- and I-tags). Accuracy is computed at the token level by comparing predicted token labels to true token labels. At the entity level, standard CoNNL[8] precision, recall, and f1 metrics are computed using the library `seqeval`.[9] An entity is marked as true positive only if its boundaries are correctly determined (the first and the last token that form the entity) and labeled with the right entity type. The following formulas are used for computations:

$$\begin{aligned} Precision &= \frac{True\ Positives}{Total\ Predicted\ Positives} \\ Recall &= \frac{True\ Positives}{Total\ Actual\ Positives} \\ F1 &= \frac{2 * (Precision * Recall)}{(Precision + Recall)} \\ Accuracy &= \frac{True\ Predicted\ Token\ Labels}{All\ Token\ Labels} \end{aligned} \tag{4}$$

Table 4
Settings of masked language models.

Model	# Hidden Layers	# Hidden Units	# Attention Heads	Vocabulary Size
mBERT	12	768	12	119,547
BERTurk	12	768	12	32,000
XLMR-b	12	768	12	250,002
XLMR-l	24	1024	16	250,002

[8] The Conference on Natural Language Learning that is organized by SIGNLL (ACL's Special Interest Group on Natural Language Learning).
[9] https://pypi.org/project/seqeval/

For instance, consider a scenario where the sentence "Ali dün Güzelbahçe Müzesi'ni ziyaret etti." (*Yesterday, Ali visited Güzelbahçe Museum.*) is labeled with a NER sequence that is slightly different than its true label sequence as shown below. Here, only the first token is properly labeled since its boundary and entity type are correctly determined. However, the third and fourth words are not treated as tokens that form a single entity. Although their entity types are correctly identified, they are treated as two separate named entities. Therefore, the precision and recall scores are computed as 1/3 and 1/2, respectively. Despite having boundary errors, the accuracy over the predicted sequence is computed as 5/6 since five out of six tokens are labeled correctly.

True NER Label Sequence:
`[B-PERSON, O, B-LOCATION, I-LOCATION, O, O]`
⇒ `PERSON:[(1,1)], LOCATION:[(3,4)]`
Predicted NER Label Sequence:
`[B-PERSON, O, B-LOCATION, B-LOCATION, O, O]`
⇒ `PERSON[(1,1)], LOCATION:[(3,3), (4,4)]`

5. Results and discussion

The literature on Turkish NER studies has benefited from BiLSTM neural networks and transformer-based networks on different settings. Although a dataset is common to all these studies, they have various similar and dissimilar design considerations, parameter settings, and initializations in their architectures. In this work, we not only provide the most comprehensive performance evaluations that compare two different architectures on the same experimental setup but also report the impact of some design choices that have not been explored before in these architectures. Following an ablation study, we present our findings and quantify the strength of effect of each design consideration in focus on different architectures. Finally, we contribute to the literature by introducing a transformed-based model with a CRF layer at the top and demonstrate that this model outperforms the current state-of-art Turkish NER studies.

5.1. Experiments on BiLSTM-CRF networks

We built several BiLSTM models using different configurations and conducted experiments to assess the impact of a single design parameter

at each turn. Table 5 presents the performance scores of some models (in increasing order of f1 scores) on validation and test sets, respectively. We choose these models since they reflect the general tendency of varying parameters between models.

It is our observation that previous Turkish research has spent tremendous effort to find the best way of forming input word embeddings and explored different combinations of vectors that represent word tokens from different perspectives. Character and morphological embeddings were shown to have a positive effect on the performance of BiLSTM networks (Güngör et al., 2019; Akkaya & Can, 2020). However, to the best of our knowledge, no previous research has measured the contribution of subword information in encoding. A character sequence of a word is often longer than its subword sequence and longer sequences present significant modeling challenges for Seq2Seq models. Moreover, subword representations result in modest vocabulary size and have the potential to form basis for robust feature representations once accompanied with character-based representations. Thus, we argue that unexplored effect of subword representations on BiLSTM performance is worth studying.

The comparison between Model 1 and Model 3 shows a slight performance increase of 0.51% on the validation set and 1.17% on the test set when subword embeddings are used instead of word embeddings. The observed increase might stem from a shorter vocabulary size (reduced data sparsity) which circumvents the problem of out-of-vocabulary words up to a level. However, the rise is not that significant as we expected. This might be due to the presence of character embeddings which might efficiently encode information carried in suffixes and thus surpass advantages of subword units. Although average scores over 5 different runs are reported, one particular reason for performance differences might be the fact that randomly initialized word or subword embeddings converge differently at each run. Additionally, performance differences are observed on training and validation sets over different epochs as shown in Fig. 4.

Our second set of experiments are designed to assess the contribution of word embeddings, in particular initializations of word embeddings, to tagging performance. Model 1 that uses randomly initialized word vectors achieves an f1 measure of 85.75% on our validation set. Although, Hur and FastText pretrained embeddings both contribute to that performance with 1.46% and 3.69% increases respectively, the highest addition of 5.82% comes from Huaw embeddings. We also observe similar performance improvements on the test set as shown in Table 5. The results that we obtain from Models 4, 6, and 8 as compared to Model 1 motivate the need for pretrained embeddings as a good starting point. Moreover, a bigger dataset and larger word embeddings result in substantial improvements on measured performance. Moving from Hur embeddings (Model 4) to Huaw word embeddings (Model 8) provides an increase of 4.36% on the validation set and 3.92% on the test set. An increase of 2.23% on validation and 2.07% on test sets are observed when we shift from Hur embeddings (Model 4) to FastText embeddings (Model 6) and we relate this change to dimensional differences between these embeddings. Huaw embeddings (Model 8) improve f1 scores by 2.13% on validation and 1.85% on test sets as compared to FastText embeddings (Model 6). This is possibly due to different methods utilized in learning representations since FastText treats each word as a composition of character ngrams, whereas Huaw embeddings are obtained by treating each word as a single token.

Our final set of experiments, in line with previous research, also confirms that the use of a CRF layer on top of the underlying architecture significantly improves f1 measures both on validation and test sets (Models 2 and 8). In a sequence labeling task, it is not surprising to see a positive effect of modeling dependencies between consecutive input tokens. In addition, f1 score obtained from Model 5 by utilizing a CRF layer is higher than that obtained from Model 2 where character embeddings are used. However, in our experiments we do not measure any notable improvements once morphological embeddings are incorporated (Models 7 and 8) and this does not support the findings of Güngör et al., 2019 where a higher improvement is obtained with the addition of morphological information. As shown in Fig. 5, performances of these two models are very similar during training and validation. One particular reason for this divergence might be the fact that a character-only model was not used with a larger dimension in that study as we used here.

This experimental study reveals that the learning method used to obtain word embeddings matters, so do their dimensions; which is supported by the work of Melamud, McClosky, Patwardhan, and Bansal, 2016. The importance of embeddings is also mentioned in the work of Ma and Hovy, 2016 where GloVe embeddings lead to the highest performance on English. Finally, our experiments strengthen the effectiveness of utilizing character embeddings as demonstrated in the work of Kuru et al., 2016. This finding might be attributed to morphological information that may possibly be carried out by individual characters.

5.2. Experiments on transformer-based networks

In the literature, there exists a limited amount of work where a transformer-based language model was applied to Turkish NER task and these works were shown to outperform the state-of-the-art results obtained by BiLSTM networks (as shown in Table 2). However, there are a few other transformer-based large language models whose performances have not been reported for Turkish tagging task. Additionally, to our

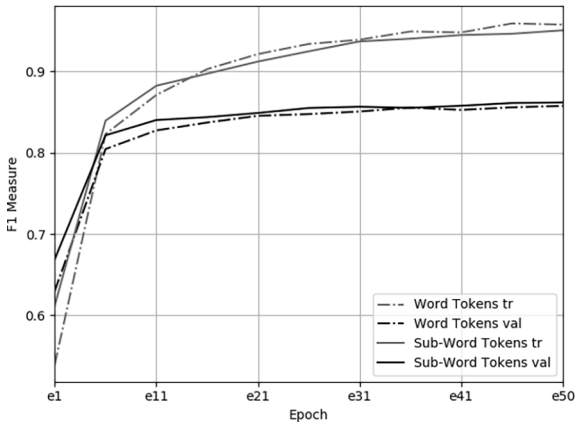

Fig. 4. Performance comparisons of word and subword embeddings during training and validation.

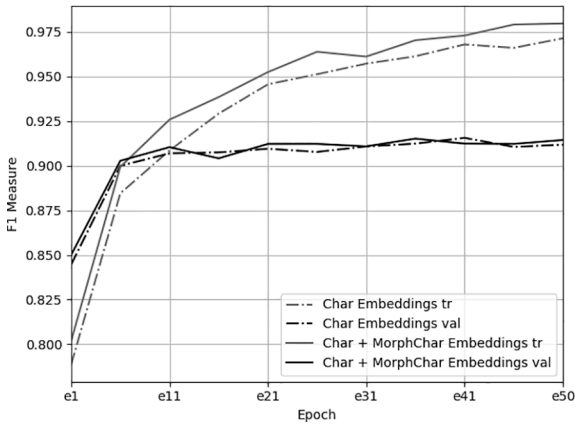

Fig. 5. Performance comparisons of character and character-morphological embeddings during training and validation.

Table 6
Performance scores of transformer-based models on our validation and test sets.

Model #	Model Description	Valid\Test	F1	Precision	Recall	Accuracy	Trn. Time
1	mBERT-CRF	Valid	92.65	91.90	93.42	98.92	03:47:00
		Test	92.35	91.54	93.17	98.90	
2	mBERT	Valid	92.73	92.07	93.40	98.96	01:48:25
		Test	92.59	91.74	93.45	98.94	
3	XLMR-b-CRF	Valid	93.29	92.65	93.95	99.02	03:52:08
		Test	93.89	93.10	94.69	99.11	
4	XLMR-b	Valid	93.29	92.61	93.99	99.05	01:55:42
		Test	94.01	93.10	94.93	99.15	
5	XLMR-l*	Valid	94.56	93.90	95.24	99.21	03:21:20
		Test	94.82	93.99	95.66	99.28	
6	BERTurk	Valid	94.87	94.37	95.38	99.28	01:39:42
		Test	95.75	95.41	96.10	99.41	
7	BERTurk-CRF	Valid	94.90	94.48	95.33	99.28	03:44:16
		Test	95.95	95.60	96.31	99.42	

*For XLMR-large, reported training time is with a larger instance type, C5.9xlarge. All other models are trained with C5.4xlarge instance type on AWS Elastic Compute Cloud service [a1]. Results are averaged over 5 random initializations.

best knowledge, the impact of CRF on such models has not been evaluated before. Thus, our experiments on transformer-based networks are oriented around these research questions.

For this set of experiments, we trained three different networks where a multilingual cased BERT language model was used with and without a CRF layer at the top in the first architecture. Similarly, XLM-RoBERTa (XLMR) and Turkish BERT (BERTurk) language models were utilized along with CRF layers in the second and third networks, respectively. The results of these experiments on validation and test sets are reported in Table 6.

Our first observation reveals that mBERT (Models 1 and 2) performs comparably poorer than other models and BERTurk (Models 6 and 7) obtains highest f1 scores on both datasets. The results are as we expected for XLMR models; a higher performance is obtained once XLMR large (Model 5) is used rather than XLMR base (Model 4) with less number of hidden units and layers. Comparing multilingual models, we find that XLMR-b performs better than mBERT (with 0.56% and 1.42% increases on validation and test sets), and XLMR-l enhances this improvement by an additional rise of 1.27% on validation set and 0.81% on test set. In the literature, XLMR was shown to improve NER benchmarks in multiple languages (Conneau et al., 2019), and our findings provide additional support by showing that Turkish is better represented with XLMR than mBERT. Some of this difference might be attributed to better subword token production by XLMR. Moreover, XLMR tokenizer produces more similar tokens to BERTurk tokenizer and uses a larger model settings and corpus. We argue that, due to these reasons, it achieves a closer performance to BERTurk (0.93% difference on test set) than mBERT (2.23% difference on test set).

It is quite surprising to measure lower performances in mBERT (Model 1) and XLMR (Model 3) models when it is accompanied with a CRF layer. However, CRF on BERTurk (Model 7) slightly improves f1 scores with an increase of 0.03% on valid and 0.2% on test sets (as compared to Model 6), respectively. Although none of these performance changes are significant, our results correlate with previous studies that perform well without using a CRF layer (Devlin et al., 2018; Conneau et al., 2019).

Our final and most important finding is that all transformer-based models outperform BiLSTM models on Turkish NER task as shown in Fig. 6. The comparison between Model 8 from Table 5 and Model 1 from Table 6 shows an increase of at least 1.08% on validation set and 0.54% on test set. The improvement achieved with BERTurk-CRF (Model 7 in Table 6) is at most 3.33% on validation set and 4.11% on test set, respectively[10].

To observe how the outputs of these models may differ, see the example below. The sentence *"The festival begins with Rahsan Apay's concert on Cemal Resit Rey Concert Hall."* is annotated differently where the best transformer-based model outperforms the best BiLSTM model, by predicting the correct entity sequence. In this sentence, the concert hall is named after a person (Cemal Resit Rey), but the BiLSTM model fails to identify that name as part of the location (the concert hall name).

Word	**Word-Char-BiLSTM-CRF**	**BERTurk-CRF**
Festival	O	O
,	O	O
Rahşan	B-PERSON	B-PERSON
Apay'ın	I-PERSON	I-PERSON
Cemal	B-PERSON	B-LOCATION
Reşit	I-PERSON	I-LOCATION
Rey	B-LOCATION	I-LOCATION
Konser	I-LOCATION	I-LOCATION
Salonu'nda	I-LOCATION	I-LOCATION
verecegi	O	O
konserle	O	O
başlıyor	O	O
.	O	O

This study also supports the observations of previous NER studies on other morphologically rich languages (i.e., Czech, Hungarian and Finnish) where BiLSTM-CRF networks were studied. For instance, a recent study revealed that neural models obtain the highest performance once character and morphological embeddings are used along with word embeddings (Güngör et al., 2019). Literature has also showed that transformer-based networks without a CRF layer outperform LSTM-based networks in morphologically rich languages (Arkhipov et al., 2019; Virtanen et al., 2019; Ács, 2021). In these studies, different performances were measured with respect to the language model being used. For instance, XLM-Roberta was observed to perform worse than BERT but better than mBERT in Hungarian. Similarly, BERT was shown to outperform mBERT in Finnish. The addition of a CRF layer improved the model performance in Slavic BERT, which was trained for four languages including Czech. Lastly, some studies in Finnish and Hungarian demonstrated the positive impact of training data size on model performances as we observed in our study.

Our work has a number of limitations that need to be considered in future research. First, our findings are limited to formal and well-written texts, and are not generalizable to ill-formed texts such as social media content. Moreover, the nature of short texts such as search queries is

[10] One particular disadvantage of transformer-based models is the observed slow inference time (between 98–211 s) as compared to BiLSTM models (between 8–13 s).

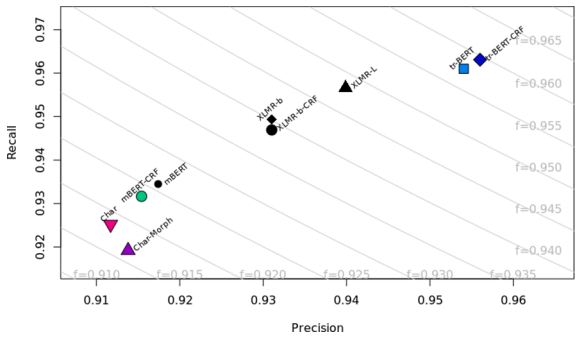

Fig. 6. Performance comparisons of BiLSTM and transformer-based models on test set.

different than longer texts that we address in this work. Therefore, our models might exhibit different performances under these circumstances. Another limitation is that our insights are specifically for Turkish and might not be applicable to other morphologically rich languages. In this work, only three NER labels are considered but there is a shift towards focusing on more entity types in order to respond to current needs of language processing applications. An in-depth study on a dataset with more NER labels might overcome this limitation. Not utilizing character and morphological embeddings in transformer-based networks and assessing their impact on model performances might be considered as a limitation of our study as well.

6. Conclusion and future work

Recent years have witnessed a surge of interests in Turkish named entity recognition. This study presents our empirical evaluations of recent neural sequence tagging models on Turkish NER task by providing a high-level comparison of different model settings and design considerations. Our results provide insights into the importance of word representations (i.e., character, morphological, subword, and word embeddings) and their initializations (i.e., random or pretrained initializations) in BiLSTM networks. Our experiments also include a comprehensive evaluation of neural architectures that utilize popular multilingual transformer-based language models on Turkish entity tagging. Their comparisons with BiLSTM models reveal their superior performance on the evaluation set and highlight the positive impact of transfer learning. In this work, we also propose a state-of-the-art transformer-based architecture with a CRF layer that achieves the highest f-measure of 94.90% and 95.95% on the validation and test sets, respectively.

We have a number of future directions which we believe will stimulate other research studies in Turkish. We first plan to aggregate character and morphological embeddings with transformer-based language models and assess their impact on the overall performance. A recent work exploring the use of character embeddings in transformer-based networks achieved state-of-the art results in German and Dutch (Yu, Bohnet, & Poesio, 2020). Therefore, it would be interesting to explore the use of these embeddings in transformer-based models in Turkish. We also intend to study other kinds of embedding methods, especially those that are not yet studied in Turkish such as Flair (Akbik et al., 2018), a slightly older LSTM-based method; and LUKE (Yamada, Asai, Shindo, Takeda, & Matsumoto, 2020), a very recent NER-specific transformer-based method. As these methods worked very well in other languages, experimenting with them would be an important addition to our research agenda; however it is worth noting that transformer methods are currently more popular, thus will likely be prioritized.

We also plan to develop new subword tokenizers such as a tokenizer that returns morphemes attached to a word as produced by a morphological analyzer. We believe that a transformer-based model trained with subwords that correctly represent the morphemes would better capture the meanings of Turkish morphological suffixes. We observe that larger the training dataset used in transformer-based models, the more likely their tokenizers can capture Turkish morphemes. Thus, we argue that training a model with a large dataset that consists of correct morpheme-based subwords would significantly improve its performance by reducing noise generated through incorrectly formed subwords.

Finally, we believe that training with larger dataset would improve the performance of all Turkish NER models. Thus, developing a larger and more extensive dataset (including new entity types in additon to person, location, and organization) for Turkish named entity recognition would be beneficial for all researchers working in the field. More specifically, one of our motivations in this study is to semantically annotate large amounts of digital news content, which requires tagging entity types that span a wider scope. As a step in this direction, we would like to explore performances of different NER tagging models on this new dataset and present the insights that we gain from this study.

[a] All training was done using machines from Amazon Web Services with instance type C5.4xlarge (https://aws.amazon.com/ec2/instance-types/c5/). These machines have Intel Xeon Scalable Processors (Cascade Lake) with a sustained all-core Turbo CPU frequency of 3.6 GHz, 16 vCPUs and 32 GiB of memory.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Authors would like to thank Kemal Oflazer, Onur Güngör and Tunga Güngör for their assistance in obtaining the Turkish NER dataset.

References

Ács, J., Lévai, D., Nemeskey, D.M., & Kornai, A. (2021). Evaluating contextualized language models for hungarian.arXiv preprint arXiv:2102.10848.
Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1638–1649).
Akdemir, A., Shibuya, T., & Güngör, T. (2020). Hierarchical multi task learning with subword contextual embeddings for languages with rich morphology. arXiv preprint arXiv:2004.12247.
Akkaya, E. K., & Can, B. (2020). Transfer learning for turkish named entity recognition on noisy text. *Natural Language Engineering*, 1–30.
Al-Nabki, M. W., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2020). Improving named entity recognition in noisy user-generated text with local distance neighbor feature. *Neurocomputing, 382*, 1–11.
Arkhipov, M., Trofimova, M., Kuratov, Y., & Sorokin, A. (2019). Tuning multilingual transformers for named entity recognition on slavic languages. BSNLP' 2019, 89.
BERTurk, 2020. Berturk:bert models for turkish. url:https://github.com/stefan-it/turkish-bert. Accessed: 2020-03-11.
Chen, L., & Moschitti, A. (2018). Learning to progressively recognize new named entities with sequence to sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2181–2191). Santa Fe, New Mexico, USA: Association for Computational Linguistics.
Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (pp. 167–176).
Chieu, H.L. & Ng, H.T. (2003). Named entity recognition with a maximum entropy approach, in: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, Association for Computational Linguistics, USA. p. 160–163. url:https://doi.org/10.3115/1119176.1119199, doi:10.3115/1119176.1119199.
Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics, 4*, 357–370.
Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research, 12*, 2493–2537.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

Demir, H., & Özgür, A. (2014). Improving named entity recognition for morphologically rich languages using word embeddings. In *2014 13th International Conference on Machine Learning and Applications* (pp. 117–122). IEEE.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.

Ekbal, A. & Bandyopadhyay, S. (2008). Bengali named entity recognition using support vector machine, in: Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages.

Eken, B., & Tantug, A. C. (2015). Recognizing named entities in turkish tweets. In *Proceedings of the Fourth International Conference on Software Engineering and Applications*.

Ertopcu, B., Kanburoglu, A. B., Topsakal, O., Acikgoz, O., Gurkan, A. T., Ozenc, B., Cam, I., Avar, B., Ercan, G., & Yildiz, O. T. (2017). A new approach for named entity recognition. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 474–479).

Finkel, J.R. & Manning, C.D. (2009). Joint parsing and named entity recognition, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, USA. p. 326–334.

Firth, J.R. (1957). A synopsis of linguistic theory, 1930–1955. Studies in linguistic analysis.

Grave, É., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Güngör, O., Güngör, T., & Üsküdarli, S. (2019). The effect of morphology in named entity recognition with sequence tagging. *Natural Language Engineering, 25*, 147–169. https://doi.org/10.1017/S1351324918000281

Güneş, A., & Tantuğ, A.C. (2018). Turkish named entity recognition with deep learning, in: 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1–4.

Hakkani-Tür, D. Z. (2000). *Statistical language modeling for agglutinative languages*. Unpublished doctoral thesis. Ankara, Turkey: Bilkent University, Department of Computer Engineering.

Harris, Z. S. (1954). Distributional structure. *Word, 10*, 146–162.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*, 1735–1780.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991.

Jurafsky, D., & Martin, J.H. (2018). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Draft for 3rd ed.

Kucuk, D., Arici, N., & Kucuk, D. (2017). Named entity recognition in turkish: Approaches and issues. In *Natural Language Processing and Information Systems* (pp. 176–181). Springer International Publishing.

Kucuk, D., & Yazici, A. (2010). A hybrid named entity recognizer for turkish with applications to different text genres. In *Computer and Information Sciences* (pp. 113–116). Netherlands: Springer.

Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 66–71).

Kuru, O., Can, O.A., & Yuret, D. (2016). CharNER: Character-Level Named Entity Recognition. Coling, 911–921url:http://www.aclweb.org/anthology/C16-1087.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

Lample, G. & Conneau, A. (2019). Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291.

Lin, X., Peng, H., & Liu, B. (2006). Chinese named entity recognition using support vector machines. In *2006 International Conference on Machine Learning and Cybernetics* (pp. 4216–4220).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Luong, T., Socher, R., & Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* (pp. 104–113). Sofia, Bulgaria: Association for Computational Linguistics.

Ma, X. & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354.

Marrero, M., Urbano, J., Sanchez-Cuadrado, S., Morato, J., & Gomez-Berbis, J. M. (2013). Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces, 35*, 482–489.

Martin, L., Muller, B., Suárez, P.J.O., Dupont, Y., Romary, L., de la Clergerie, É.V., Seddah, D., & Sagot, B. (2019). Camembert: a tasty french language model. arXiv preprint arXiv:1911.03894.

Melamud, O., McClosky, D., Patwardhan, S., & Bansal, M. (2016). The role of context types and dimensionality in learning word embeddings. arXiv preprint arXiv: 1601.00893.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Mollá, D., van Zaanen, M., & Smith, D. (2006). Named entity recognition for question answering, in: Proceedings of the Australasian Language Technology Workshop 2006, pp. 51–58.

Oflazer, K., Gocmen, E., Bozsahin, C., 1994. An outline of Turkish morphology. Technical Report. Technical Report TU-LANGUAGE, NATO Science Division SfS III, Brussels.

Okur, E., Demir, H., & Ozgur, A. (2016). Named entity recognition on twitter for turkish using semi-supervised learning with word embeddings. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*.

Paliouras, G., Karkaletsis, V., Petasis, G., & Spyropoulos, C. D. (2000). Learning decision trees for named-entity recognition and classification. In *ECAI Workshop on Machine*. Learning for Information Extraction.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Saju, C. J., & Shaja, A. S. (2017). A survey on efficient extraction of named entities from new domains using big data analytics. In *Proceedings of the Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*.

dos Santos, C.N., Guimarães, V. (2015). Boosting named entity recognition with neural character embeddings. CoRR abs/1505.05008.

Sker, G. A., & Eryigit, G. (2012). Initial explorations on using CRFs for Turkish named entity recognition. In *Proceedings of COLING 2012, The COLING 2012 Organizing Committee, Mumbai, India* (pp. 2459–2474). url:https://www.aclweb.org/anthology/C12-1150.

Souza, F., Nogueira, R., & Lotufo, R. (2019). Portuguese named entity recognition using bert-crf. arXiv preprint arXiv:1909.10649.

Tatar, S., & Cicekli, I. (2011). Automatic rule learning exploiting morphological features for named entity recognition in turkish. *Journal of Information Science, 37*, 137–151.

Tür, G., Hakkani-Tür, D., & Oflazer, K. (2003). A statistical information extraction system for turkish. *Natural Language Engineering, 9*, 181–210.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need, in: Advances in neural information processing systems, pp. 5998–6008.

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., & Pyysalo, S. (2019). Multilingual is not enough: Bert for finnish. arXiv preprint arXiv: 1912.07076.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. ArXiv abs/1910.03771.

Yamada, I., Asai, A., Shindo, H., Takeda, H. & Matsumoto, Y. (2020). LUKE: Deep contextualized entity representations with entity-aware self-attention. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online. pp. 6442–6454. url:https://www.aclweb.org/anthology/2020.emnlp-main.523, doi: 10.18653/v1/2020.emnlp-main.523.

Yao, L., Sun, C., Li, S., Wang, X., & Wang, X. (2009). Crf-based active learning for chinese named entity recognition. In *2009 IEEE International Conference on Systems, Man and Cybernetics* (pp. 176–185).

Yeniterzi, R. (2011). Exploiting morphology in Turkish named entity recognition system, in: Proceedings of the ACL 2011 Student Session, Association for Computational Linguistics, Portland, OR, USA. pp. 105–110. url:https://www.aclweb.org/anthology/P11-3019.

Yu, J., Bohnet, B., & Poesio, M. (2020). Named entity recognition as dependency parsing, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online. pp. 6470–6476. url: https://www.aclweb.org/anthology/2020.acl-main.577, doi:10.18653/v1/2020.acl-main.577.

Zhang, Y. & Yang, J. (2019). Chinese NER Using Lattice LSTM, 1554–1564. doi: 10.18653/v1/p18-1144, arXiv:arXiv:1805.02023v4.

Zhou, G., & Su, J. (2002). Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 473–480).

Zirikly, A. & Diab, M. (2015). Named entity recognition for Arabic social media. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 176–185.

Çelikkaya, G., Torunoğlu, D., & Eryiğit, G. (2013). Named entity recognition on real data: A preliminary investigation for turkish. In *2013 7th International Conference on Application of Information and Communication Technologies* (pp. 1–5).