Neural Computing and Applications (2024) 36:8371–8382
https://doi.org/10.1007/s00521-024-09532-1

ORIGINAL ARTICLE

# Application of BiLSTM-CRF model with different embeddings for product name extraction in unstructured Turkish text

Serdar Arslan[1]

Received: 27 April 2023 / Accepted: 22 January 2024 / Published online: 21 February 2024

## Abstract

Named entity recognition (NER) plays a pivotal role in Natural Language Processing by identifying and classifying entities within textual data. While NER methodologies have seen significant advancements, driven by pretrained word embeddings and deep neural networks, the majority of these studies have focused on text with well-defined grammar and structure. A significant research gap exists concerning NER in informal or unstructured text, where traditional grammar rules and sentence structure are absent. This research addresses this crucial gap by focusing on the detection of product names within unstructured Turkish text. To accomplish this, we propose a deep learning-based NER model which combines a Bidirectional Long Short-Term Memory (BiLSTM) architecture with a Conditional Random Field (CRF) layer, further enhanced by FastText embeddings. To comprehensively evaluate and compare our model's performance, we explore different embedding approaches, including Word2Vec and Glove, in conjunction with the Bidirectional Long Short-Term Memory and Conditional Random Field (BiLSTM-CRF) model. Furthermore, we conduct comparisons against BERT to assess the efficacy of our approach. Our experimentation utilizes a Turkish e-commerce dataset gathered from the internet, where traditional grammatical and structural rules may not apply. The BiLSTM-CRF model with FastText embeddings achieved an F1 score value of 57.40%, a precision value of 55.78%, and a recall value of 59.12%. These results indicate promising performance in outperforming other baseline techniques. This research contributes to the field of NER by addressing the unique challenges posed by unstructured Turkish text and opens avenues for improved entity recognition in informal language settings, with potential applications across various domains.

**Keywords** BERT · BiLSTM-CRF · Deep learning · FastText · Named entity recognition

## 1 Introduction

Information extraction, among many other applications, heavily relies on named entity recognition (NER). The majority of prior research on NER relies on extracting general information to identify Person (PER), Location (LOC), Organization (ORG), and Time (TIM) [1, 2]. However, to the best of our knowledge, there have been few studies available on extracting product information, even though it appears that it's significant and beneficial for information extraction in the area of e-commerce systems.

Due to the nature of these e-commerce systems, products are titled in a short and unstructured way [3]. Therefore, product name extraction or tagging from unstructured texts has a number of challenges. First of all, in unstructured text, there is no formal or well-defined sentence structure. Moreover, there are no punctuation rules in this sentence, and it might not use standard grammar or sentence construction rules. Secondly, as there is no set format for producing a product title, different titles or descriptions for the same item are possible because product titles are generally ad hoc pieces of information. Another issue is that product titles have uncommon words since they do not all contain the same word shape, for example, some of them may contain numbers or have rare words.

There are a number of data sets available for named entity recognition; however, they include structured text related to news, Wikipedia entries, or some other blog

✉ Serdar Arslan
sarslan@cankaya.edu.tr

[1] Computer Engineering Department, Cankaya University, 06790 Etimesgut, Ankara, Turkey

entries [4–8]. For noisy data, there are also some data sets collected from Twitter and some other social media resources [7, 9]. However, for the product name extraction, there are only a few e-commerce datasets, which are also mostly in English.

Turkish is a complicated morphological language with extensive features. The term "morphological language" refers to a category of language in which the smallest units of meaning in a language, known as morphemes, are combined to generate words [10]. These morphemes can be put together in a variety of ways to make new words or change the meaning of ones that already exist. In a morphological language, words often have multiple forms, depending on their grammatical function in a sentence. The use of morphemes in Turkish allows for a great deal of flexibility in word formation and can lead to complex word structures and rich systems of inflection, where a single word can convey a variety of meanings through changes in its morphological form. Therefore, many computing tasks cannot be completed by relying solely on word forms.

The problem addressed in this research is the NER task within the context of informal or unstructured Turkish text [10, 11]. Specifically, the challenge lies in identifying and classifying product names in text that lacks traditional grammar rules, sentence structure, and may not conform to standard punctuation and language conventions. This problem arises due to the prevalence of such unstructured text in various online contexts, particularly in e-commerce.

In the literature, there are a few studies that try to apply NER methodologies to structured Turkish text. In addition to this, applying the NER approaches to unstructured Turkish text is very challenging problem [12]. This paper addresses the research gap by focusing on the specific challenges of NER in informal Turkish text, with a primary emphasis on the extraction of product names. Our key research question is as follows:

Research Question: How can the Named Entity Recognition (NER) task be effectively addressed in the context of unstructured Turkish text, particularly for the extraction of product names, and how does the proposed deep learning-based NER model, incorporating Bidirectional Long Short-Term Memory and Conditional Random Field (BiLSTM-CRF) and FastText embeddings, compare with state-of-the-art language models like BERT, as well as traditional baseline NER models?

To answer this question, we propose a novel deep learning model that combines BiLSTM architecture with a CRF layer, further enhanced by FastText embeddings. Additionally, we conduct extensive experiments on a Turkish e-commerce dataset, comparing our model's performance not only with traditional NER models but also with advanced models like BERT. This research aims to provide insights into effective NER solutions for informal Turkish text and contributes to the broader field of NLP and entity recognition. To the best of our knowledge, this is the first deep learning model for product entity recognition using unstructured Turkish text.

The contributions of our work can be listed as follows:

- This work tackles a significant research gap in NER by focusing on the specific challenges presented by informal text. While existing NER models excel in structured text, they often struggle to perform effectively in unstructured contexts. This study acknowledges this gap and proposes a solution tailored to informal Turkish text.
- A new deep learning-based model for unstructured Turkish text is proposed. This model combines a Bidirectional Long Short-Term Memory architecture with a Conditional Random Field layer, bolstered by FastText embeddings. This innovative approach is designed to excel in the absence of traditional sentence structure and grammar.
- The study conducts a thorough performance evaluation of the proposed model. It not only presents the results of the Bidirectional Long Short-Term Memory and Conditional Random Field model with FastText embeddings but also explores alternative embedding approaches, including Word2Vec and Glove. This comprehensive evaluation demonstrates the effectiveness of the proposed model in addressing the research problem.
- To provide a comprehensive assessment, the proposed model's performance is compared with state-of-the-art language model (BERT). This comparison allows for a deeper understanding of the model's relative strengths and capabilities in handling unstructured Turkish text. In addition to the BERT comparison, the research also compares the proposed model's performance against baseline NER models. This comparative analysis showcases the advantages and potential of the proposed method over existing techniques.
- The research employs a Turkish e-commerce dataset collected from the internet, which simulates real-world conditions where informal text is prevalent. By focusing on product name extraction, this work has practical applications in improving search functionality and product recommendation systems in e-commerce platforms.

Section 2 reviews recent studies related to NER. The proposed deep BiLSTM-CRF model for unstructured text is described in Sect. 3. The experimental results of the algorithms are compared and discussed in Sect. 4. Our concluding remarks and possible future work directions are presented in the last section.

## 2 Related works

In the literature, named entity recognition is applied to various domains. Earlier named entity recognition models have used statistical methods and rule-based models. [13] used decision trees in their studies. [14] proposed a semi-supervised sequential labeling approach using statistical methods. In [15], maximum entropy classifiers are used for NER. Conditional random fields are used for NER in [16]. The study proposed in [17] uses statistical methods for entity extraction and recognition.

Nowadays, with recent and rapid development in artificial intelligence, machine learning models and deep neural networks (DNN) have been widely used in NER and NLP. In [18] authors use a BiLSTM-CRF architecture for sequence labeling problem. Similar to this work, in [19] an end-to-end BiLSTM-CRF architecture is proposed for the same task. In [20] Skip-Gram based embedding approach is utilized with BiLSTM-CRF for English text. Seq2Seq model is used in [21] for NER and compared with BiLSTM model. Contextual string embeddings are used to train LSTM model in [22].

NER in Turkish is a popular and also open research area [23]. Recent studies for Turkish named entity recognition have also utilized neural networks [10]. In [24], authors propose a NER model for Turkish legal texts with a custom-made corpus using BiLSTM and CRF. A neural network based model which employs a semi-supervised learning approach is presented in [25]. The work used a CRF layer on top of the decoder and concatenated word, character, and morphological embeddings as encoder inputs in [26] for Turkish language. The authors presented a LSTM based approach with stacked layers of varied depths while combining word embeddings and writing style embeddings (such as all uppercase letters or sentence case letters) as input representations in [27].

For unstructured or noisy text, there are very few studies in the literature. Çelikkaya et al. [28] presented a NER model for real-world data, which is based on a CRF model. They used morphological and lexical features of noisy Turkish text. The authors adopt a baseline approach using CRF and leverage morphological and lexical features influenced by prior work. They apply this approach to the forum data, speech data, and Twitter datasets, testing different preprocessing scenarios, such as normalization and capitalization.

Another CRF-based model is presented in [29], which employs optional distance-based matching. The authors explore two main approaches for addressing the performance drop in informal texts, namely adapting systems to the characteristics of informal texts or adapting data to suit existing systems. They propose a specialized NER system for tweets without normalization, achieving a 64% F-measure using fundamental features such as word prefixes and suffixes, capitalization, apostrophe information, and gazetteers. The study also emphasizes data preparation techniques, including asciifying datasets and gazetteers with minimal normalization. Furthermore, distance-based matching using the Levenshtein distance algorithm is employed for gazetteer look-up features, and future work aims to enhance these techniques. The paper highlights the importance of tailoring NER systems to the unique properties of informal text domains like tweets, offering valuable insights for NLP applications in this context.

In [30], the authors used morphological and lexical features to utilize the CRF model on Turkish news dataset and also Turkish tweets. The research involves extensive feature engineering to enhance performance on both well-formed texts and user-generated content, introducing new datasets from the Web 2.0 domain and expanding the coverage of named entity types. The approach achieves a promising exact match F1 score of 92% on Turkish news articles and approximately 65% on Web 2.0 datasets. While results are satisfactory for well-formed texts, further research is needed to improve recognition on non-canonical social media content, especially in the case of lowercase proper nouns.

In [31], a rule-based approach has been modeled for Turkish tweets. The study conducts experiments, adapting a rule-based recognition system to better suit Twitter language by relaxing capitalization constraints and expanding lexical resources with diacritics. Additionally, a simplistic tweet normalization scheme is introduced to assess its impact on NER. The findings provide insights into the complexities of NER in Turkish tweets and the effects of tweet normalization, suggesting desirable features for a tailored NER system.

A semi-supervised learning approach based on neural networks is proposed in [32] for Turkish tweets. The study highlights the potential of using in-domain data for unsupervised learning of word embeddings, making it adaptable for morphologically rich languages beyond Turkish.

A BiLSTM-CRF model with different types of embeddings (character, character n-gram, morphological, and orthographic character embeddings) is presented in [11]. The paper explores NER in Turkish noisy text using deep neural networks and transfer learning, as an alternative to rule-based or statistical methods. They utilized another layer to their model which is trained using both noisy and formal text simultaneously. The study tackles the challenges posed by sparse orthography, user style dependencies, and the morphologically rich structure of Turkish. It investigates various word and subword representation techniques without the use of hand-crafted features or

**Table 1** Related works on noisy Turkish data

| Study | Year | Dataset | Approach | F1 (%) |
|---|---|---|---|---|
| Celikkaya [28] | 2013 | Tweets, Forum, News [28] | CRF by using morphological features and gazetteers | 19.28 |
| Kucuk [31] | 2014 | Tweets [4, 28] | A rule-based system with two adaptations: relaxing the capitalization constraint and diacritics-based expansion of the system's lexical resources | 46.93 |
| Eken [29] | 2015 | Tweets [28] | CRF by using ASCII conversion of data sets and gazetteers and also applying a normalization | 46.97 |
| Okur [32] | 2016 | News [33], Tweets [34] | An unsupervised learning method for word embedding generation with a neural network trained on labeled data | 48.96 |
| Seker [30] | 2017 | Tweets, News [28, 35] | A CRF-based Turkish NER model to cover additional entity types and leveraging morphological information | 67.96 |
| Akkaya [11] | 2021 | Tweets [28, 30] | BiLSTM-CRF architecture by incorporating an additional CRF layer which is trained simultaneously on a larger (formal) text and a noisy text | 67.39 |

external resources. Table 1 summarizes these studies and their F1 score results on noisy data.

## 3 Proposed work

We introduced an effective, systematic, and practical approach based on a Bidirectional Long Short-Term Memory and Conditional Random Field deep learning model for product entity detection in unstructured text. This method encompasses data scraping, preprocessing, data annotation, feature engineering, deep learning model construction, training, and evaluation phases. Figure 1 provides an overview of our proposed method, outlining the steps involved in detecting entities in unstructured text.

The deep learning model comprises three main layers as shown in Fig. 2: the Embedding layer, the BiLSTM layer,

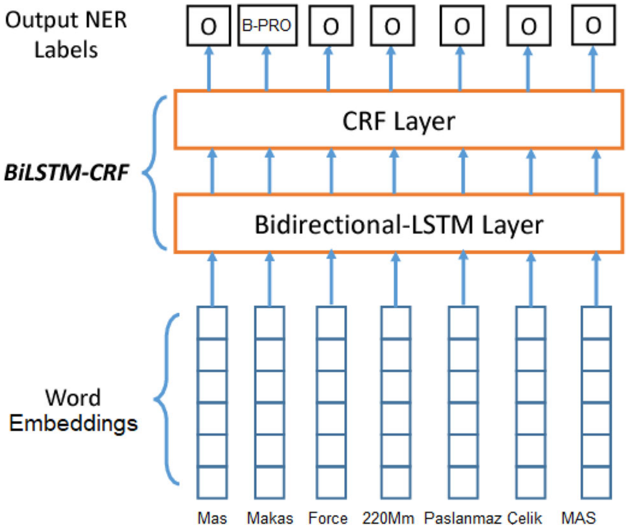
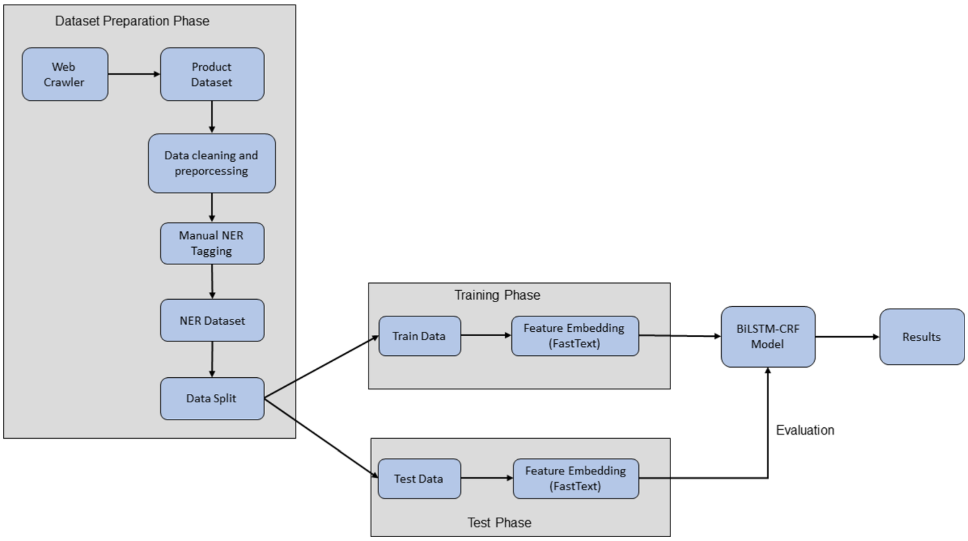


**Fig. 2** The structure of BiLSTM-CRF model

**Fig. 1** Overall structure of proposed model

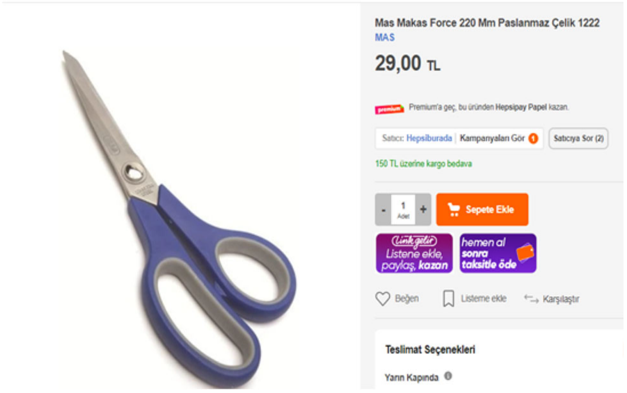


and the CRF layer. Within the Embedding layer, unstructured text undergoes tokenization, and each token is transformed into numerical form using an embedding technique. This work employs two distinct embedding techniques: FastText and BERT-based embeddings. The vector representations of unstructured text are subsequently input into the acrshortbilstm model. The output of the acrshortbilstm layer is then utilized as input for the CRF layer, which serves the purpose of classifying named entities. To facilitate comparison, two distinct BiLSTM-CRF models were developed for each of the embedding techniques.

## 3.1 Data collection and annotation

The dataset is collected from Turkish e-commerce web site [36]. The product titles are first extracted and cleaned using preprocessing tools. After cleaning the titles, manual labeling performed on each row of the dataset which contains 1167 rows. The snapshot of e-commerce site is shown in Fig. 3. From this product page, the text *"Mas Makas Force 220 Mm Paslanmaz Çelik 1222 MAS"* is extracted and then the product name in this text is labeled manually which is *Makas (Scissors*. There is only one type of entity, namely PRODUCT(PRO), in this work and the labels used to annotate the product entities are in IOB2 format. These format contains Inside, Outside and Begin tags and shown in Table 2.

As an example, the product title *"Mas Makas Force 220 Mm Paslanmaz Çelik 1222 MAS"* first cleaned and numeric data is removed. After this process, we obtain the clean product title as *"Mas Makas Force Mm Paslanmaz Çelik MAS"*. For this text we have the tag representations shown in Table 3.



Mas Makas Force 220 Mm Paslanmaz Çelik 1222
MAS
29,00 TL

Premium'a geç, bu üründen Hepsipay Papel kazan.

Satıcı: Hepsiburada | Kampanyaları Gör 1 | Satıcıya Sor (2)

150 TL üzerine kargo bedava

premium
- 1 Adet +
Sepete Ekle
Link gelir Listene ekle, paylaş, kazan
hemen al sonra taksitle öde
Beğen
Listeme ekle
Karşılaştır

Teslimat Seçenekleri

Yarın Kapında

**Fig. 3** Sample product page taken from e-commerce website [36]

**Table 2** IOB2 Format Tags for NER

| Tag | Description |
|---|---|
| B-PRO | The beginning of product name |
| I-PRO | Part of product name |
| O | Not named entity |

**Table 3** Annotations of sample text using IOB2 Format

| Word (Turkish) | Word (English) | Tag |
|---|---|---|
| Mas | Brand | O |
| Makas | Scissors | B-PRO |
| Force | Brand | O |
| Mm | Millimeter | O |
| Paslanmaz | Stainless | O |
| Çelik | Steel | O |
| MAS | Brand | O |

## 3.2 Embedding layer

In natural language processing (NLP), the word embedding approach is used to represent words as numerical vectors. This is accomplished by giving each word in a corpus (collection of text) a vector with a fixed size. These vectors are usually generated using a neural network-based method, where the network is trained on huge amount of text data and the word vectors are learned by optimizing a specific objective function. The generated vectors store various aspects of each word's meaning, including its context, grammar, and semantics.

The ability to manipulate words mathematically by performing operations like vector addition and subtraction on them makes it feasible to gain insightful knowledge about the relationships between words. In several NLP tasks, such as sentiment analysis, named entity recognition, and machine translation, word embeddings have shown to be tremendously helpful. Additionally, they are utilized in a number of well-known deep learning models for NLP, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). In short, word embedding is simply defined as mapping words to high-dimensional feature vectors.

In the literature, there are popular studies which uses different approaches to represent words in numeric format such as FastText, Glove, and Word2Vec.

FastText is a recent word embedding library developed by Facebook. The n-gram feature of the phrase is used as an additional feature to input by FastText model in addition to the word representation for each word in the sentence. This feature makes FastText as a ideal technique for generating words' vector representations of morphologically complex language such as Turkish. In this work, we used FastText embeddings as morphological embedding layer and compared it with other two state-of-the-art embedding techniques, Glove and Word2Vec.

In addition to FastText, we also used BERT model as word embedding approach. BERT (Bidirectional Encoder Representations from Transformers), released in late 2018. Traditional word embeddings generate a fixed-length vector representation for each word in a vocabulary, based on its co-occurrence statistics with other words in a large corpus of text. In contrast, BERT uses masked language modeling (MLM) to train a deep bidirectional transformer network on a huge corpus of text to provide embeddings for complete sentences or text sequences. BERT randomly masks some of the input tokens during training, and the model has been assigned with predicting the masked tokens based on the context.

As a result, embeddings that accurately capture the context-dependent meaning of each word in the sentence are created, enabling the model to acquire a deep representation of the relationships between words in a sentence. By layering a task-specific layer on top of the pretrained BERT model and optimizing the entire network on a task-specific dataset, the embeddings can be fine-tuned on a downstream NLP task, such as sentiment analysis or named entity recognition, once the model was successfully trained.

In this work, we use BERT to extract features, namely word embedding vectors, from unstructured text data. BERT can express tokenized words as corresponding word embeddings. BERT is better at handling unlabeled data and the word vector expressed by BERT includes context information in addition to its own information. We utilized a pretrained language mode, namely Turkish BERT (BERTurk) for this purpose.

## 3.3 Deep learning models

### 3.3.1 BILSTM

Recurrent neural network (RNN) that can handle the vanishing gradient problem is known as Long Short-Term Memory (LSTM) network. LSTM is also better at maintaining long-range connections and understanding the connection between values at the start and end of a sequence. By modifying a gating structure of a traditional RNN model, the LSTM model can learn or retain a longer data sequence. Therefore, LSTM has three gates: input, forget, and hidden. An LSTM unit with these three gates and a memory cell forms a layer of neural network neurons, and each neuron has a hidden layer and a current state. Figure 4 shows the LSTM cell's structural layout.

The forget gate is used to specify whether or not certain data will be kept. This preservation is accomplished using the following formula;

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

where $x_t$ denotes input at time $t$, $h_{t-1}$ denotes the output of previous cell, and $\sigma$ is a sigmoid function. If a forget gate outputs 1 (one), the information is stored in the cell state. The sigmoid function creates a vector in the following step. New possible values are stored in this vector. The updated values are specified by input gates, and the vector $C_t^{'}$ is updated with possible new values. This new vector is evaluated with the following formulas;

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$
$$C_t^{'} = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

Now cell's old state $C_{t-1}$ is updated to new cell state $C_t$.

$$C_t = f_t * C_{t-1} + i_t * C_t^{'}$$

Eventually, we select the network's output regarding on the cell state. This selection process is carried out by using the following formulas;

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

A special type of LSTM network which is used generally for natural language processing is called Bidirectional Long Short-Term Memory. In BiLSTM, there are two different LSTM networks in order to represent the input in both direction (backward and forward). Therefore, representing the input in both directions of the sequence, it is an effective tool for modeling the sequential relationships between words and phrases. In BiLSTM, two hidden states are introduced, one for accessing the context of previous input $\overleftarrow{h_t}$, and another one for accessing the context of the next input $\overrightarrow{h_t}$. Therefore, the formula for the hidden state of BiLSTM can be defined as:

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t}$$

In summary, BiLSTM reverses the direction of information flow by adding one extra LSTM layer. It simply means that in the additional LSTM layer, the input sequence flows backward. The outputs from the two LSTM layers are then combined in a variety of ways, including average, sum, multiplication, and concatenation. BiLSTM process the input sequence of $S = w_1, w_2, \ldots, w_n$ sequentially. Here $w_i$ denotes the $i$th word of the sentence $S$. The embedding of

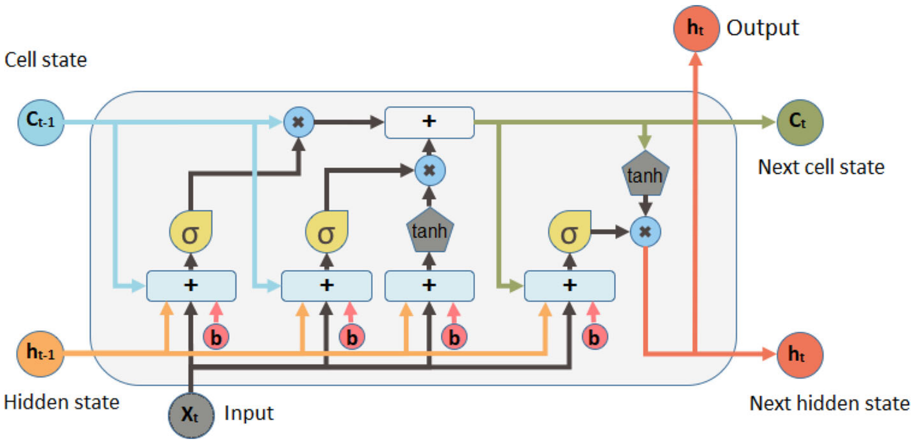


**Fig. 4** LSTM cell structure

each word $w$ is supplied to BiLSTM model in order to produce the concatenated output $X$.

### 3.3.2 CRF

A Conditional Random Field is a type of Markov random fields and they can be used to predict a specific label or tag using its neighbor labels with some consistency. CRF is generally utilized to predicate these adjacent labels in sequential data. In this work, CRF layer is utilized as top hidden layer of BiLSTM network to concatenate the last hidden states from the underlying network. The CRF layer tries to predict the sequence of labels $Y = y_0, y_1, \ldots, y_n$ of a given sentence $X = x_0, x_1, \ldots, x_n$by using following equation:

$$p(y \mid x) = e^{\mathrm{Score}(x,y)} / \sum_{y'} e^{\mathrm{Score}(x,y')}$$

In this equation, $p(y \mid x)$ denotes the conditional probability, and Score is computed using the following equation:

$$Score(x, y) = \sum_{i=0}^{T} A_{y_i, y_{i+1}} + \sum_{i=1}^{T} P_{i, y_i}$$

where $A_{y_i, y_{i+1}}$ represents probability of transition from label $i$ to label $j$, and $P_{i,j}$ denotes the score of the $j$th label of the word $i$th. This $P_{i,j}$ is represented as matrix and it is the output of BiLSTM network. With CRF model, the main goal is to maximize the log of conditional probability $log(p(y \mid x))$.

### 3.3.3 BiLSTM-CRF

This architecture has three layers: an embedding layer, a Bidirectional LSTM layer, and a dropout layer, just like the core BiLSTM model. However, it also has the following two extra parts:

- TimeDistributed(Dense) Layer: This layer applies a dense layer to each time step of the BiLSTM output sequence, allowing the network to learn features at each time step.
- CRF Layer: This layer applies a CRF algorithm to the output of the TimeDistributed(Dense) layer, which models the dependencies between adjacent output labels and helps to ensure that the predicted label sequence is globally optimal.

Overall, the BiLSTM-CRF model has been proven to be superior to many other state-of-the-art models in sequence labeling tasks such named entity recognition and part-of-speech tagging.

## 3.4 Evaluation

We employ Precision, Recall, and F1 score as key metrics to assess the effectiveness of our proposed model. Precision gauges the model's capability to accurately identify relevant entities, while Recall evaluates its ability to capture all relevant entities within a dataset. The F-score, on the other hand, serves as the harmonic mean of Precision and Recall, offering a balanced measure of the model's overall performance. Detailed calculations for these metrics are provided below.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$F1 = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} = \frac{2 * \mathrm{TP}}{2 * \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

where True Positive (TP) represents entities correctly recognized by the model and that align with annotated entities. False Positive (FP) pertains to entities erroneously identified by the model, which do not align with annotated entities. Conversely, False Negative (FN) encompasses annotated entities that the model fails to recognize.

## 4 Experiments

In the first set of experiments, we compared performance results of different BiLSTM-CRF models for each embedding approach. We implemented our architecture in Python with TensorFlow and Keras. The dataset contains 1167 labeled rows and split into train and test sets with %80-%20 ratio. All experiments were conducted on a single personal computer equipped with a single NVIDIA RTX 3080 GPU boasting 32 GB of memory.

In the course of our research, we implemented a systematic strategy aimed at fine-tuning the hyperparameters of our deep learning model, a crucial element in attaining exceptional performance levels. To facilitate this process, we employed a methodology referred to as grid search. This approach provided us with the means to comprehensively explore a broad spectrum of hyperparameter combinations, systematically adjusting key parameters such as learning rates, batch sizes, and model architecture configurations. This diligent exploration led to the identification of highly promising hyperparameter configurations that notably bolstered the model's effectiveness.

The adoption of grid search underscored our unwavering commitment to meticulous parameter optimization, driving us toward the attainment of the most favorable model settings and, ultimately, yielding exceptional outcomes in our study. Hence, the hyperparameter values for BiLSTM model implementation were optimized for all set of experiments as follows: batch size 128; epoch 50; learning rate 0.002; LSTM hidden size 50; and LSTM dropout rate 0.1.

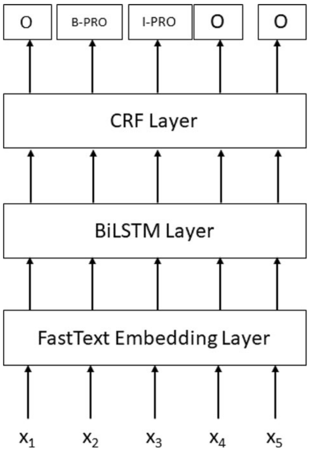


**Fig. 5** BiLSTM-CRF model with FastText embedding

**Table 4** Performance results of proposed work with FastText embedding and different dimensions

| Model Name | Embedding | Dimension | F1 | Precision | Recall |
|---|---|---|---|---|---|
| BiLSTM-CRF | FastText | 50 | 53.69 | 52.71 | 54.65 |
| BiLSTM-CRF | FastText | 300 | 57.40 | 55.78 | 59.12 |

First of all, we conducted test on BiLSTM-CRF with FastText embedding layer (Fig. 5). In the embedding layer we used two different dimension number, respectively, 50 and 300. The results in Table 4 show that there is a little performance gain using 300 dimensional embedding vector.

In order to show the performance of the proposed work, we utilized BiLSTM-CRF with two other embeddings: Glove and Word2Vec. The performance results are shown in Table 5. When compared to alternative word embedding strategies, using the FastText word embedding model significantly improves model performance.

Moreover, we designed second set of experiments to compare the proposed approach with BERT embedding (Fig. 6) and the results are shown in Table 6. This experiment results reveal that using BERT for embedding layer shows very close performance to proposed model. The F1 scores for each model are shown in Fig. 7 and accuracy results for each model are shown in Fig. 8. The experiments reveal that using morphological embedding for unstructured Turkish text outperforms word based embedding.

In addition to these set of experiments, we conducted another set of experiments to compare our model performance with those of two baseline methods. In the first baseline method BERT transformer model is utilized with CRF layer on top of this model (BERT-CRF) (Fig. 9). In the second baseline model, a standard BiLSTM model is utilized without CRF layer and FastText embedding is used as embedding layer again (BiLSTM FastText) (Fig. 10).

**Table 5** Comparison of BiLSTM-CRF model with different embedding layers

| Model Name | Embedding | F1 | Precision | Recall |
|---|---|---|---|---|
| BiLSTM-CRF | FastText | 57.40 | 55.78 | 59.12 |
| BiLSTM-CRF | Glove | 50.15 | 58.62 | 51.74 |
| BiLSTM-CRF | Word2Vec | 52.05 | 50.13 | 54.06 |

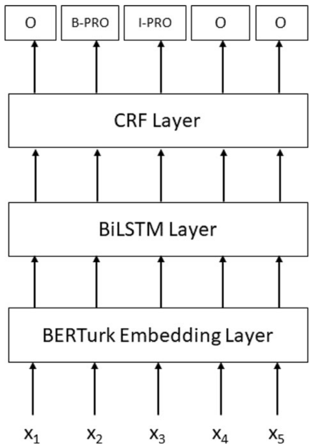


**Fig. 6** BiLSTM-CRF model with BERT embedding

**Table 6** Performance results of proposed models with FastText and BERT embedding

| Model Name | Embedding | Dimension | F1 | Precision | Recall |
|---|---|---|---|---|---|
| BiLSTM-CRF | FastText | 300 | 57.40 | 55.78 | 59.12 |
| BiLSTM-CRF | BERT | 768 | 56.04 | 53.87 | 58.39 |

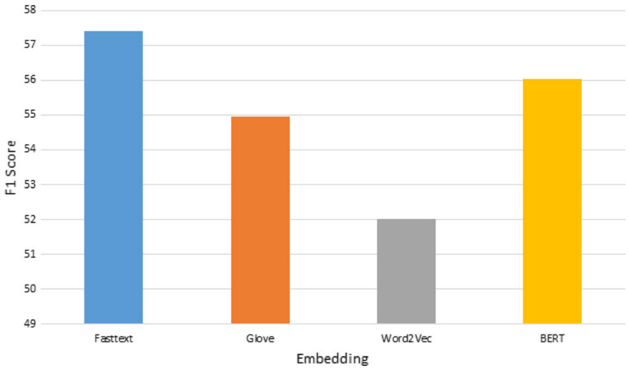


**Fig. 7** F1 results of each embedding technique

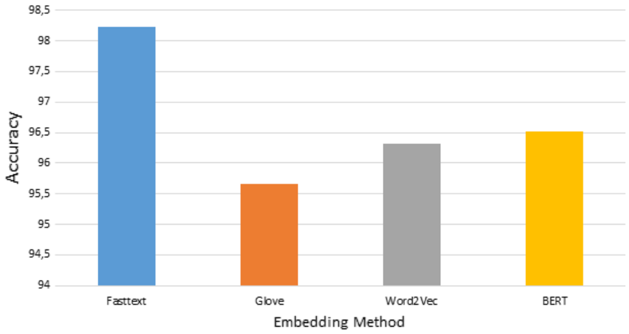


**Fig. 8** Accuracy results of each embedding technique

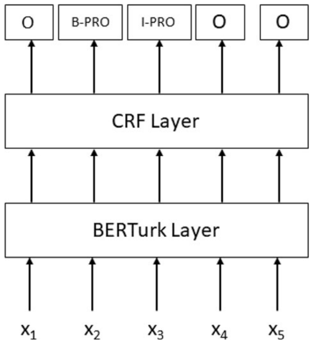


**Fig. 9** BERT-CRF model

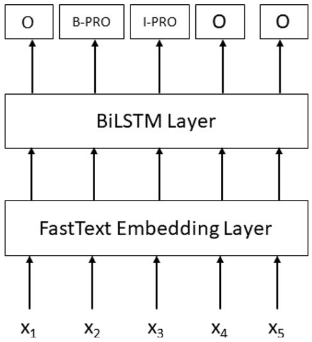


**Fig. 10** BiLSTM (without CRF layer) model with FastText embedding

**Table 7** Comparison results of proposed model and baseline models

| Model Name | Embedding | F1 | Precision | Recall |
|---|---|---|---|---|
| BiLSTM-CRF | FastText | 57.42 | 55.78 | 59.12 |
| BiLSTM | FastText | 52.08 | 50.65 | 53.58 |
| BERT-CRF | | 56.41 | 54.01 | 58.95 |

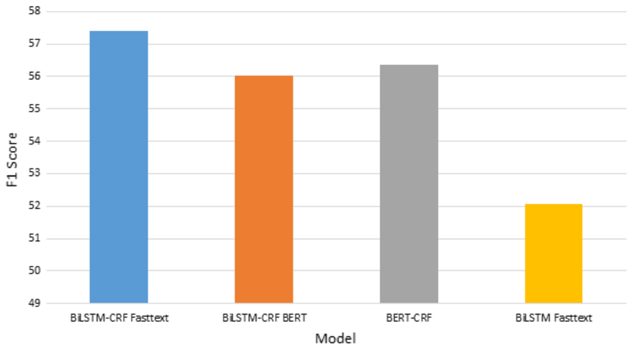


**Fig. 11** F1 results of proposed model and baseline models

The test results are summarized in Table 7 and overall F1 score and accuracy values for these model and also BiLSTM-CRF with BERT embedding is depicted in Figs. 11 and 12 for overall comparison. The results also

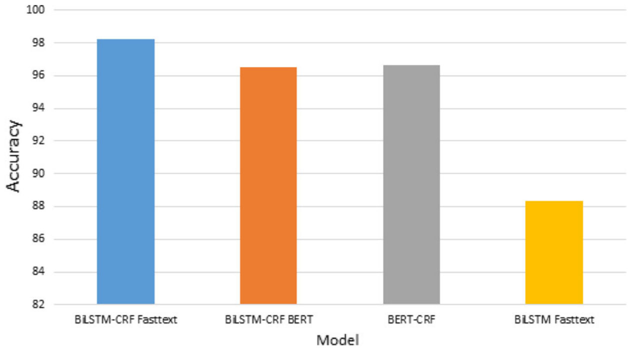


**Fig. 12** Accuracy results of proposed model and baseline models

confirm that our proposed model has better results than these baseline models. Moreover, using FastText embeddings has better results compared to BERT embeddings. FastText embeddings may be a better choice than BERT embeddings because they are generated based on character-level n-grams, which allows them to capture subword information and handle variations in spelling and morphology. This capability makes them more robust to unstructured text data than traditional word embeddings, which can be easily affected by misspellings, typos, and other errors. In contrast, BERT embeddings are contextualized and rely heavily on the overall structure and meaning of the sentence. Therefore, BERT embeddings are less robust to unstructured text data, as errors or variations in the text can disrupt the overall context and meaning of the sentence.

### 4.1 Error analysis

To examine the underlying factors contributing to misclassification by the techniques in our corpus, we compiled and synthesized the results of the experiments. To achieve this objective, we conducted a qualitative error analysis to investigate common errors in the results. Typically, product names, which are frequent, are accurately tagged. However, when they are infrequent or written with symbols and numbers, there is a possibility that they may not be tagged correctly. Another typical error occurs when product names consist of multiple words, especially if irrelevant and meaningless words are located between them.

Another error occurs when product names are shortened. For example for the product definition *Petlas 205/55 R16 91 H MultiAction PT565 Oto 4 Mevsim Lastiği ( Üretim Yılı: 2022 )* (winter tire product definition and "Oto" is shortened for "Otomobil" which means Automobil or car) "Oto" word is used to refer "Otomobil". Therefore, the expected tagging is "Otomobil 4 mevsim lastiği" (car four season tire), but the model tagged it as "Mevsim Lastiği".

These are the common errors, however, the model correctly tags frequent and single-word production names.

## 5 Conclusion

Dealing with unstructured data is a challenging task in Natural Language Processing (NLP), and there have been several contributions to improving Named Entity Recognition (NER) on this type of data.

In this study, we propose a BiLSTM-CRF model designed for unstructured Turkish text. Our methodology begins by collecting data from Turkish e-commerce websites. Subsequently, the data undergoes a meticulous cleaning process involving the removal of unwanted characters, symbols, noise (such as numbers, punctuation, and special characters), and redundant or irrelevant information.

The transformed input data is then converted into numerical representations utilizing embedding techniques. When handling unstructured textual data, it proves beneficial to employ word embeddings that exhibit resilience to text variations and errors. Our model employs diverse embedding approaches to address these variations and errors. Particularly, we utilize FastText, a widely used library developed by Facebook, to generate word embeddings based on character-level n-grams.

Given that Turkish is a language known for its morphological complexity, the utilization of FastText embeddings allows us to capture subword information. This feature equips the model to effectively manage variations in spelling and morphology. This stands in contrast to conventional word embeddings, which are generated at the word level and may falter in accommodating spelling variations or morphological distinctions among related words.

As an alternative embedding approach, we incorporate BERT embeddings to represent unstructured text. BERT embeddings have demonstrated exceptional performance across a broad spectrum of NLP tasks, even when dealing with noisy or imperfect text data. Their efficacy stems from their ability to grasp intricate contextual relationships between words within a sentence.

We proceed to compare these two embedding techniques with each other and with established traditional embedding methods, all within the framework of a BiLSTM-CRF model. The core of our model lies in the bidirectional Long Short-Term Memory architecture, which adeptly captures the context of the input sequence. Additionally, we incorporate a Conditional Random Field layer to model interdependencies among the output labels. This sophisticated architecture enables the model to predict the label of each token within the input sequence by leveraging both forward and backward contextual information.

We conducted our experiments using real-world datasets, and the results reveal compelling findings. Notably, when applied to Turkish, a language known for its morphological complexity, FastText embeddings outperform traditional embedding techniques. Furthermore, our results indicate that FastText embeddings exhibit a slight performance advantage over BERT embeddings.

In the pursuit of advancing the capabilities of product name recognition in unstructured text, several promising avenues for future research emerge. First and foremost, there is the compelling prospect of extending the model's proficiency to multilingual contexts. Given the diverse linguistic landscape of e-commerce platforms, exploring the adaptation of our model to different languages could enhance its applicability.

Another promising avenue involves the fine-tuning of the model for specific domains within the e-commerce sector, such as electronics, fashion, or food. Tailoring the model to recognize domain-specific product names can lead to heightened accuracy and relevance in recommendations.

To further refine our approach, advanced data preprocessing techniques warrant exploration. This includes strategies for handling misspelled words and text that departs from standard grammar and syntax, ensuring the model's robustness in the face of diverse textual idiosyncrasies. Moreover, the implementation of active learning strategies holds potential for reducing the manual annotation burden associated with data collection. By selecting the most informative instances for manual labeling, we can not only enhance model performance but also alleviate annotation costs.

These future research directions collectively pave the way for further innovation and advancement in the realm of product name recognition, with broad-reaching implications for e-commerce and related fields.

**Funding** Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). The research received no external funding.

**Data availability** The datasets used and/or analyzed during the current study and codes are available from the corresponding author on reasonable request.

## Declarations

**Conflicts of interest** The authors declare that they have no competing interests.

## References

1. Marrero M, Urbano J, Sánchez-Cuadrado S, Morato J, Gómez-Berbís JM (2013) Named entity recognition: fallacies, challenges and opportunities. Comput Stand Interfaces 35(5):482–489. https://doi.org/10.1016/j.csi.2012.09.004
2. Goyal A, Gupta V, Kumar M (2018) Recent named entity recognition and classification techniques: a systematic review. Comput Sci Rev 29:21–43. https://doi.org/10.1016/j.cosrev.2018.06.001
3. Shah SAA, Ali Masood M, Yasin A (2022) Dark web: E-commerce information extraction based on name entity recognition using bidirectional-LSTM. IEEE Access 10:99633–99645. https://doi.org/10.1109/ACCESS.2022.3206539
4. Kucuk D, Jacquet G, Steinberger R (2014) Named entity recognition on Turkish tweets. In: Proceedings of the ninth international conference on language resources and evaluation (LREC14), European Language Resources Association (ELRA), Reykjavik, pp 450–454
5. Akmal M, Romadhony A (2020) Corpus development for Indonesian product named entity recognition using semi-supervised approach. In: 2020 international conference on data science and its applications (ICoDSA), pp 1–5. https://doi.org/10.1109/ICoDSA50139.2020.9212879
6. Ding N, Xu G, Chen Y, Wang X, Han X, Xie P, Zheng H-T, Liu Z (2021) Few-NERD: a few-shot named entity recognition dataset
7. Malmasi S, Fang A, Fetahu B, Kar S, Rokhlenko O (2022) SemEval-2022 task 11: multilingual complex named entity recognition (MultiCoNER). In: Proceedings of the 16th international workshop on semantic evaluation (SemEval-2022), Association for Computational Linguistics, Seattle, pp 1412–1437. https://doi.org/10.18653/v1/2022.semeval-1.196. https://aclanthology.org/2022.semeval-1.196
8. Ruokolainen T, Kauppinen P, Silfverberg M, Linden K (2019) A finish news corpus for named entity recognition. Lang Resour Eval 54(1):247–272. https://doi.org/10.1007/s10579-019-09471-7
9. Zhang H, Hennig L, Alt C, Hu C, Meng Y, Wang C (2020) Bootstrapping named entity recognition in E-commerce with positive unlabeled learning. In: Proceedings of the 3rd Workshop on e-Commerce and NLP. Association for Computational Linguistics, Seattle, WA, pp 1–6. https://doi.org/10.18653/v1/2020.ecnlp-1.1. https://aclanthology.org/2020.ecnlp-1.1
10. Aras G, Makaroğlu D, Demir S, Cakir A (2021) An evaluation of recent neural sequence tagging models in Turkish named entity recognition. Expert Syst Appl 182:115049. https://doi.org/10.1016/j.eswa.2021.115049
11. Kağan Akkaya E, Can B (2021) Transfer learning for Turkish named entity recognition on noisy text. Nat Lang Eng 27(1):35–64. https://doi.org/10.1017/S1351324919000627
12. Ozcelik O, Toraman C (2022) Named entity recognition in Turkish: a comparative study with detailed error analysis. Inf

Process Manag 59(6):103065. https://doi.org/10.1016/j.ipm.2022.103065

13. Paliouras G, Karkaletsis V, Petasis G, Spyropoulos CD (2000) Learning decision trees for named-entity recognition and classification. In: ECAI workshop on machine learning for information extraction
14. Suzuki J, Isozaki H (2008) Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In: Proceedings of ACL-08: HLT. Association for Computational Linguistics, Columbus, pp 65–673. https://aclanthology.org/P08-1076
15. Chieu HL, Ng HT (2003) Named entity recognition with a maximum entropy approach. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, pp 160–163. https://aclanthology.org/W03-0423
16. Finkel JR, Manning CD (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, Boulder, pp 326–334. https://aclanthology.org/N09-1037
17. Wu Y, Zhao J, Xu B (2003) Chinese named entity recognition combining a statistical model with human knowledge. In: Proceedings of the ACL 2003 workshop on multilingual and mixed-language named entity recognition. MultiNER '03. Association for Computational Linguistics, vol 15, pp 65–72. https://doi.org/10.3115/1119384.1119393
18. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. Preprint arXiv:1508.01991
19. Ma X, 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. Association for Computational Linguistics, Berlin, vol 1, no Long Papers, pp 1064–1074. https://doi.org/10.18653/v1/P16-1101. https://aclanthology.org/P16-1101
20. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, San Diego, pp 260–270. https://doi.org/10.18653/v1/N16-1030. https://aclanthology.org/N16-1030
21. 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. Association for Computational Linguistics, Santa Fe, pp 2181–2191. https://aclanthology.org/C18-1185
22. Akbik A, Blythe D, Vollgraf R (2018) Contextual string embeddings for sequence labeling. In: Proceedings of the 27th international conference on computational linguistics. Association for Computational Linguistics, Santa Fe, pp 1638–1649. https://aclanthology.org/C18-1139
23. Küçük D, Arıcı N, Küçük D (2017) Named entity recognition in Turkish: Approaches and issues. In: Frasincar F, Ittoo A, Nguyen LM, Métais E (eds) Natural language processing and information systems. Springer, Cham, pp 176–181
24. Çetindağ C, Yazıcıoğlu B, Koç A (2022) Named-entity recognition in Turkish legal texts. Nat Lang Eng. https://doi.org/10.1017/S1351324922000304
25. 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. https://doi.org/10.1109/ICMLA.2014.24
26. Güngör O, Güngör T, Üsküdarli S (2018) The effect of morphology in named entity recognition with sequence tagging. Nat Lang Eng 25:147–169
27. Güneş A, TantuG AC (2018) Turkish named entity recognition with deep learning. In: 2018 26th signal processing and communications applications conference (SIU), pp 1–4. https://doi.org/10.1109/SIU.2018.8404500
28. 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. https://doi.org/10.1109/ICAICT.2013.6722801
29. Eken B, Tantuğ A (2015) Recognizing named entities in Turkish tweets. vol 5, pp 155–162. https://doi.org/10.5121/csit.2015.50213
30. Seker GA, Eryiğit G (2017) Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content. Sem Web 8:625–642
31. Küçük D, Steinberger R (2014) Experiments to improve named entity recognition on Turkish tweets. In: Proceedings of the 5th workshop on language analysis for social media (LASM). Association for Computational Linguistics, Gothenburg, pp 71–78. https://doi.org/10.3115/v1/W14-1309. https://aclanthology.org/W14-1309
32. Okur E, Demir H, Özgür 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 (LREC'16). European Language Resources Association (ELRA), Portorož, pp 549–555. https://aclanthology.org/L16-1087
33. Sak H, Güngör T, Saraçlar M (2011) Resources for Turkish morphological processing. Lang Resour Eval 45(2):249–261. https://doi.org/10.1007/s10579-010-9128-6
34. Sezer B, Sezer T (2013) TS corpus: Herkes için Türkçe derlem. In: Proceedings of the 27th national linguistics conference (March), pp 217–225
35. Tür G, Hakkani-Tür D, Oflazer K (2003) A statistical information extraction system for Turkish. Nat Lang Eng 9(2):181–210. https://doi.org/10.1017/S135132490200284X
36. Hepsiburada: online e-commerce site. http://www.hepsiburada.com

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.