Turkish Named Entity Recognition: A Survey and Comparative Analysis

Anonymous ACL submission

Abstract

Named entity recognition is a challenging task that has been widely studied in English. Although there are some efforts for named entity recognition in Turkish language, the reported results are limited to particular datasets and models. Moreover, there is a lack of comparative analysis for named entity recognition in Turkish. In this study, we contribute to the literature in three folds. First, we provide an up-to-date short survey on Turkish named entity recognition studies. Second, we compare state-of-the-art named entity recognition models on various Turkish datasets that we can access to. Lastly, we analyze a set of linguistic processing steps that would affect the performance of Turkish named entity recognition.

1 Introduction

Named entity recognition (NER) is an essential subtask of information extraction, which finds the predetermined named entity classes in a text. NER is frequently used as a key component in several NLP applications; such as Information Retrieval (Mandl and Womser-Hacker, 2005), Question-Answering (Pizzato et al., 2006), Machine Translation (Babych and Hartley, 2003), Automatic Text Summarization (Nobata et al., 2002).

Although NER is widely studied in many languages as English (Yadav and Bethard, 2018), Chinese (Ma et al., 2020), and Arabic (Shaalan, 2014); NER is still a challenging task for agglutinative and morphologically rich languages, such as Turkish.

Turkish NER studies report high performance results on the well-written text datasets, such as news articles (Aras et al., 2020; Gunes and Tantug, 2018). On the other hand, excessive usage of social media results in noisy text data, such as tweets, including lots of spelling errors, abbreviations, semantic ambiguity, and user-generated words. This makes noisy texts difficult to analyze, which results in lower performance (Akkaya and Can, 2021) From these perspectives, models and datasets need to be analyzed by comparing several datasets with various models to understand their generalization capability.

In this study, to give a big picture of up-to-date studies, we first provide a short survey on Turkish named entity recognition. We then compare state-of-the-art named entity recognition models to analyze them various in Turkish datasets. We lastly provide a linguistic analysis on the performance of models.

Our contributions can be summarized in three folds. First, we provide an up-to-date short survey on Turkish named entity recognition studies. Second, we compare state-of-the-art named entity recognition models on various Turkish datasets that we can access to. Lastly, we analyze a set of linguistic processing steps that would affect the performance of Turkish named entity recognition.

The rest of the paper is organized as follows. We give a detailed review of Turkish NER studies in Section 2. We report our experimental results in Section 3. We provide a brief discussion on main insights and Turkish-specific challenges in Section 4. We conclude the study in the last section.

2 A Short Survey on Turkish NER

In Turkish NER studies, employing morphological and syntactic features is commonly observed widely to increase the performance of the experiments. A brief summary of the Turkish NER studies with datasets and the F1 scores are presented in Table 1. According to our observations, we divide the related work subsections considering methods through studies, which are rule-based, machine learning, neural networks and transformer-based studies.

2.1 Rule-based Studies

Earliler traditional NER studies are composed of rule-based studies that need linguistic experts to

Study	Approach	Data	F1
Aras et al. (2020)	BERTurk-CRF	News	95.95
Gunes and Tantug (2018)	Deep-BiLSTM	News	93.69
Güngör et al. (2018)	CRF	News	93.37
Şeker and Eryiğit (2012)	CRF	News	91.94
Tür et al. (2003)	HMM	News	91.56
Akdemir and Güngör (2019)	CRF	News	89.89
Yeniterzi (2011)	CRF	News	88.94
Şeker and Eryiğit (2017)	CRF	Tw-DS	67.96
Akkaya and Can (2021)	BiLSTM-CRF	Tw-DS	67.39
Eken and Tantuğ (2015)	CRF	Tweets	63.77
Onal and Karagoz (2015)	WAN	Tw7	57.26
	WAN	Speech	71.54
Küçük and Steinberger (2014)	Rule-based	Tw7	54.81
Çelikkaya et al. (2013)	CRF	Media	91.64
	CRF	Twitter	13.88
	CRF	Speech	50.69
Yilmaz et al. (2020)	BiLSTM-CRF	TW	82.90
	BiLSTM-CRF	FB	83.90
	BiLSTM-CRF	DH	83.70

Table 1: A summary of related studies on Turkish NER.

analyze the resource and craft language dependent features. Large gazetteers and normalizers are commonly employed in these studies. Küçük and Yazici (2009) presented the first rule-based NER system on news articles, child stories, and historical texts with 78.7%, 69.3%, and 55.3% F1 scores, respectively. They did not use the capitalization and punctuation rules to make the system robust for noisy texts.

Tatar and Cicekli (2011) developed an automatic rule learning system and reported F1 score of 91.08% on the TurkIE dataset manually tagged on terrorism using both online and printed newspapers. Küçük and Yazıcı (2012) proposed the first hybrid Turkish named entity recognizer. In this study they improved the model to learn from annotated data when available. They achieved a hybrid system employing the high success rate of rule based system on the dataset used in Küçük and Yazici (2009).

Küçük and Steinberger (2014) implemented a rule-based system on tweets by adapting its rules to fit the datasets better by relaxing capitalization constraints and by diacritics-based expansion, and they also employed a simplistic normalization scheme. They experimented on two different Turkish tweet datasets. They reported scores on the their tweet dataset, Tw-DS (Küçük and Steinberger, 2014), and Twitter dataset (Çelikkaya et al., 2013).

2.2 Machine Learning Studies

Machine learning based studies in Turkish NER widely consists of CRF method, which is a probabilistic model to label sequence. In these studies feature engineering is done to craft inputs of CRF.

Tür et al. (2003) presented the first Turkish NER study based on Hidden Markov Models (Rabiner and Juang, 1986), which is a statistical learning approach on a well-written dataset, News Articles (*News*). Yeniterzi (2011) provided improvement over their own baseline (Yeniterzi, 2011) by 7.6%. They implemented a CRF-based system employing roots and morphological features of words and used a morpheme-level tokenization method that represents the word as root and morphological feature states. Şeker and Eryiğit (2012) presented explorations on the usage of morphological structure as features to the CRF with some gazetteers. They reported the highest F1 scores until then with and without gazetteers.

Çelikkaya et al. (2013) prepared three new noisy Turkish datasets whose domains are Twitter, Speech-to-Text Interface, and Hardware Forum. They used the same method as in (Şeker and Eryiğit, 2012) with an addition of a normalizer at the morphological processing in order to normalize noisy data. They created three different models composing different sets of features and tested them with normalization and without normalization. They reported relatively lower success rates on different noisy datasets in comoparison with the well-written text datasets News Media (*Media*).

Eken and Tantuğ (2015) creates the Tweets dataset and merge them with Twitter data (Çelikkaya et al., 2013) to train CRF and test the model on Tweets Test split. Taşpınar et al. (2017) implemented different machine learning approaches by benefiting word embeddings along with the Tweet-specific syntactic features. Şeker and Eryiğit (2017) proposed a CRF-based framework employing from morpheme level processing. They achieved 67.96% F1 score on Tw-DS, which is the re-annotated version of Çelikkaya et al. (2013). Güngör et al. (2018) employed RNN (Rumelhart et al., 1986) structure to create context vector embeddings and CRF model to predict named entities.

Akdemir and Güngör (2019) proposed a hybrid model that makes use of hand-crafted features. Dependency parsing related features together with other features is used to boost NER performance. The model is CRF-based and uses News Articles (*News*) dataset (Tür et al., 2003).

2.3 Neural Network studies

Neural Networks methods are exemplified as BiLSTM and BiLSTM-CRF models in Turkish NER

studies, which are sequentially process the words. Demir and Özgür (2014) implemented a semi-supervised learning approach based on neural networks using the framework of Ratinov and Roth (2009), who employs regularized averaged perceptron algorithm. They adopted a fast unsupervised method to learn continuous vector representations of the words and used them with language independent features. They improved previous state-of-the-art result by 2.26% over Şeker and Eryiğit (2012) (overall 91.85%) without using gazetteers for Turkish and Czech by 1.53% over Konkol and Konopík (2013) (overall 75.61%). Unlike previous works their system does not make use of any language dependent features; thus, it is implementable also for other morphologically rich languages like Czech.

Onal and Karagoz (2015) obtained word embeddings and used them as features to train Window Approach Network (WAN) of the SENNA, which is the NLP from Scratch framework proposed in Collobert et al. (2011). They trained the word embeddings on a large and merged unannotated text corpus (Boun Web Corpus Sak et al. (2008) and Turkish Wikipedia[1]) containing around 500M tokens, with a vocabulary of size 954K. Moreover, the NER Classifier model is learned in supervised manner on Tür et al. (2003) data set. Evaluating on six different data sets from previous studies, they improved the F1 score on Tw7 (Küçük and Steinberger, 2014) to 57.26% from 48.13%, for Speech dataset (Çelikkaya et al., 2013) to 71.54% from 50.84%.

Okur et al. (2016) utilized a semi-supervised learning approach based on neural networks where a regularized averaged multiclass perceptron is used. They employed Skip-gram model to obtain word vectors using word2vec Mikolov et al. (2013) on Boun Web Corpus, together with language independent features that are engineered to work better on informal text types. In addition, for supervised learning steps, they used News Articles (*News*), Twitter dataset (Çelikkaya et al., 2013) and Tw7 (Küçük and Steinberger, 2014). They achieved the state-of-the-art until then for Twitter dataset and Tw7 with 48.96% and 56.79% respectively.

Gunes and Tantug (2018) utilized a neural network application. They implemented RNN architecture via using BiLSTM and Deep-BiLSTM. They improved the F1 score of (Güngör et al., 2018) by 0.10%, and their best model is reported with 93.69% F1 score.

Yilmaz et al. (2020), proposed a hybrid framework, and created informal datasets from Twitter, Facebook[2], and Forum Website Donanimhaber[3]. This dataset is annotated by three annotators with 16 NER tags and whole data set includes 1,671,665 words which is larger than the most commonly used dataset (*News*) in Turkish NER studies. They employed word embeddings, character representation, morphological features, POS tags and gazetteers to compose word representation. Cross-domain experiments were done on three datasets. The best F1 scores of their study are 83.8% for Twitter (TW), 85.3% for Facebook (FB), and 84.5 % for Forum Website (DH).

Akkaya and Can (2021) present transfer learning by adopting a deep recurrent neural network model without using any hand-crafted features. As input to BiLSTM-CRF model, different levels of word embeddings are used. One CRF model is trained on a large dataset which is the re-annotated version of News Articles (*News*) (Tür et al., 2003), and the other one is trained on a small dataset which is noisy-informal Twitter dataset. Thus, the model learns from both data set jointly, and transfer learning implemented on Tweet-DS (Şeker and Eryiğit, 2017).

2.4 Transformer-based studies

Transformer-based models capture content of the sentence and location of the each words in the sentence, which provide contextual information and long-range dependencies, based on Transformer architecture (Vaswani et al., 2017). In Turkish NER, there are limited studies on transformer-based models. Aras et al. (2020) empirically investigate the recently used neural architectures and concluded that transfer-based networks overcome the limitations of BiLSTM networks. They also proposed a transfer-based network with a CRF layer on top, which is the current state-of-the-art model on the News Articles (*News*) dataset. Our comparative analysis includes several Transformer-based NER models including English and multi-lingual language models to understand their generalization capabilities to Turkish NER, and also Turkish language models.

[1] https://tr.wikipedia.org/

[2] https://www.facebook.com/

[3] https://forum.donanimhaber.com/

Definition	News Articles		WikiANN-tr		Tweets-CG		Tweets-FG		ATIS-NER	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Word Count	444,940	47,249	155,951	75,731	78,208	8,683	78,208	8,683	43,846	7,358
Person	14,690	1,600	8,827	4,517	1,617	275	1,617	275	-	-
Location	9,763	1,116	9,547	4,850	960	144	960	144	7,165	1,208
Organization	9,158	866	7,946	4,142	1,360	168	1,360	168	-	-
Date & Time	-	-	-	-	-	-	241	36	2,064	357
Other	-	-	-	-	-	-	122	6	971	175
Sentences	19,322	3,327	19,990	9,999	7,824	855	7,824	855	4,978	890
Vocab. Size	71,007		44,011		31,476		31,476		2,569	

Table 2: Main statistics of Turkish datasets used in this study. Other includes money, percentage, code, and names.

3 Experiments

There are two main experiments in this study. First, we compare the performances of the state-of-the-art models for Turkish NER. Second, we analyze important linguistic processing steps that would affect the performance of Turkish NER.

3.1 Datasets

We can access[4] to four datasets, two of which are formal text in terms of language style, and the remaining sets are informal text having daily language. The motivation is to compare the model performances on well-written text with daily language. News Articles (Tür et al., 2003) and WikiANN-tr (Rahimi et al., 2019) are formal datasets composed of news articles and Wikipedia texts. Tweets (Eken and Tantuğ, 2015) and ATIS-NER are informal datasets composed of Turkish tweets and airline spoken queries translated from English to Turkish. We modify Tweets to have a coarse-grained version (Tweets-CG), and use the original version (Tweets-FG) as in (Eken and Tantuğ, 2015).

The dataset statistics are given in Table 2. We use these datasets since we can not access to other Turkish NER datasets (Çelikkaya et al., 2013; Şeker and Eryiğit, 2017; Yilmaz et al., 2020). Since there are several studies that report their results on Turkish NER using different datasets, we aim to use all datasets that we can access to, and examine the generalization capability of state-of-the-art models on Turkish datasets.

3.1.1 News Articles

The News Articles (*News*) dataset (Tür et al., 2003) has Turkish news articles annotated with the ENAMEX-type named entities, i.e. person, location, and organization types (PLO). The dataset includes news articles of a Turkish newspaper, Milliyet, from January 1997 to September 1998. The named entities are tagged according to IOB2 scheme. There are various modified versions of this dataset used in previous studies on Turkish NER. We use the dataset version published by Çelikkaya et al. (2013). Since the data is already processed and clean, we do not apply any cleaning steps.

3.1.2 WikiANN-tr

The *WikiANN-tr* dataset (Rahimi et al., 2019) is the Turkish subset of a multi-lingual NER dataset consisting of Wikipedia articles annotated with the ENAMEX type. However, we observe false annotations, and also Arabic and Russian sentences in WikiANN-tr. We therefore apply the following cleaning steps. We split suffixes separated by apostrophes. For instance, "Ankara'da" (translated to "at Ankara") is split to "Ankara" and "'da". Suffixes with apostrophes are mostly used when word is a named entity in Turkish. We split text according to punctuation marks.

3.1.3 ATIS-NER

The *ATIS-NER* (Airline Travel Information System) dataset includes spoken queries (utterances) annotated for the task of slot filling in conversational systems (Goo et al., 2018; Mesnil et al., 2014). Since the task is similar to NER, we adapt Turkish version of ATIS, provided by (Şahinuç et al., 2020), and refer it to as ATIS-NER. This fine-grained version has 64 slot labels (or named entities) in IOB2 format. We apply the same cleaning steps as in WikiAnn-tr. In addition, we apply the following steps. Due to domain of ATIS, there are labels related to airline codes, flight numbers, and transport types. We clean the dataset so that it can be used in Turkish NER studies. Slot labels, such as *fromloc.city_name* and *toloc.city_name*, are merged into the same entity (*city_name*). Some of slot labels have common information, so we also merge them into the same entity (e.g. *city_name*

[4]To access a dataset, we first try to download if publicly available. Otherwise, we ask the authors who used the dataset in their study.

and *airport_name* are merged into the NAME tag). In addition, *depart_date.day_name*, *depart_time.time*, *fate_amount*, and *airport_code* are tagged as DATE, TIME, MONEY, CODE respectively. Overall, we map all related slot labels to their NER tags. We remove relative labels, such as *return_date.date_relative* and *cost_relative*, and unnecessary labels such as *meal*, *economy*, *transport_type* that are not related to any ENAMEX or TIMEX tags.

3.1.4 Tweets Dataset

The *Tweets* dataset (Eken and Tantuğ, 2015) has 9,358 tweets tagged in ENAMEX, NUMEX and TIMEX types. We apply the same cleaning steps as in WikiAnn-tr. In addition, we apply the followings. We remove all duplicate tweets; decreasing the number of tweets to 8,967. We observe leaks in the test set, e.g. 21 tweets in the train set are also seen in the test set, which are removed. We remove #(hashtags), @(user-names), RT(retweet) and URLs. If a tweet contains only hashtags and usernames, it is removed from the dataset. We replace multiple repeated characters with their single equivalence, e.g. "Hello::)))" is converted to "Hello:)"

Since News Articles and WikiANN-tr are tagged in ENAMEX type, in order to be comparable, we also create a ENAMEX-tagged version, called as Tweets Coarse-Grained (*Tweets-CG*), by changing DATE, TIME, MONEY, and PERCENTAGE to the empty 'O' tag. We also use the original tagged version after removing duplicates and cleaning processes, called as Tweets Fine-Grained (*Tweets-FG*).

3.2 Evaluation Metrics

We measure the performances in terms of precision, recall, and weighted F1 score, which are standard CoNLL (Sang and De Meulder, 2003) metrics. Evaluation is done using the seqeval library (Nakayama, 2018).

We use IOB2 format in this study. I, O and B stands for inside, outside, and beginning, respectively (Ramshaw and Marcus, 1999). There are various versions of IOB tagging format, e.g. IOBES and IOB1. However, IOB2 is one of the most frequently used format in NER. In IOB2 tagging, each entity chunk starts with B-<class>, and continues with I-<class>. In this study, all datasets are in the IOB2 format, where entities are determined by grouping the tokens to form a single entity (see Table 2).

3.3 Comparative Analysis

We compare the following models for Turkish named entity recognition.

CRF Tagging A tagging model can exploit features for each input in text sequence to find outputs for each independently. Conditional Random Fields (Lafferty et al., 2001) is a probabilistic model that considers neighbor tag information jointly. We employ CRF as a tagging layer in this study.

Neural Networks Recurrent neural architectures can process text sequentially to obtain neural embeddings that represent text sequence at every step for named entity recognition (Lample et al., 2016). LSTM (Hochreiter and Schmidhuber, 1996) is a recurrent neural model that captures long-range dependencies in text with several gate structures. We employ BiLSTM (Graves and Schmidhuber, 2005), is a bi-directional LSTM model that take advantage of processing text sequence from both backward and forward. To utilize a tagging model, we employ BiLSTM-CRF (Huang et al., 2015), which employs a CRF layer above a bi-directional LSTM encoding layer. We give FastText (Bojanowski et al., 2017) word embeddings as input to the BiLSTM encoding layer.

Transformer Language Models Transformer is a deep learning-based architecture that uses self-attention for each token over all tokens (Vaswani et al., 2017). Text sequence is processed bi-directionally as in BiLSTM, but with self-attention that keeps positional embeddings. There is a family of Transformer-based language models, mostly pretrained using English data. We use two major models in this family, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). BERT is a bi-directional language model with the tasks of masked language model and next sentence prediction. RoBERTa is built on the BERT architecture with a diverse corpora. The task of next sentence prediction is also removed in RoBERTa.

BERT and RoBERTa are pretrained for English data. To understand their genelization capability, we fine-tune them for the downstream task of Turkish NER by adding a softmax layer with the cross-entropy loss function. We use the CLS sentence embeddings of the last layer as input to the softmax layer. We use the bert-base-cased model (BERT-b-c) with 12 layers, a hidden size of 768, and 12

heads; and the roberta-base model (RoBERTa-b) with 12 layers, a hidden size of 768, and 12 heads.

Multi-lingual Language Models Instead of language-specific models, multiple languages can be incorporated into pretraining phase. The advantage is that low-resource languages can benefit from high-resource languages by using shared vocabulary and semantic relatedness. We fine-tune mBERT and XLM-R that include Turkish during their pre-training phases. mBERT (Devlin et al., 2019) is built on the BERT architecture, but using multilingual data covering 100 languages. XLM-R (Conneau et al., 2020) is built on the RoBERTa architecture, but using multilingual data covering 100 languages. XLM-R removes next sentence prediction, and has more data than mBERT in training.

Turkish Language Models We fine-tune BERTurk, DistilBERTurk, ConvBERTurk, elecTRa, which are pre-trained by using only Turkish text (Schweter, 2020). BERTurk re-trains the BERT architecture for Turkish data. We employ the BERTurk versions trained with vocabulary sizes of 32k and 128k. The distilBERTurk model is a distilled version of BERTurk with a smaller training data. ConvBERTurk is based on ConvBERT (Jiang et al., 2020), but using a modified training procedure and Turkish data. The elecTRa model is based on ELECTRA (Clark et al., 2020), using Turkish data.

3.3.1 Experimental Setup

We report results on the original train and test splits. Original split is given to compare with the studies used only original split, however original split has only a train-test split which might yield to randomness in the training model. We therefore merge original train and test splits, and then apply 10-fold leave-one-out cross-validation in order to avoid potential annotator-dependent effects (Larson et al., 2019), and get reliable average scores over multiple splits.

We use the pre-trained Turkish word embeddings provided by FastText[5] (Bojanowski et al., 2017) word embeddings to feed LSTMs. We use Tensorflow[6] library for BiLSTM and BiLSTM-CRF. We design BiLSTM as they composed of 50 units in forward and backward layers. After concatenation of forward and backward layers outputs there is a dense layer consisting 100 perceptrons. Finally, at the top of the structure we use sigmoid activation function. LSTMs are trained with no dropout, a learning rate of 5e-3, a bath size of 16, and 20 epochs.

We use Simple Transformers[7] library to train Transformer-based language models. We train them with the following hyper-parameters; learning rate is 5e-5, number of epochs is 10, and batch size is 16. The training process is done with a GeForce RTX 2080 Ti.

3.3.2 Experimental Results

The results using different models on Turkish datasets with their original train and test splits are given in Table 3, where results are weighted averaged over 10 time repeated training since the models are stochastic, i.e. each run can generate a different result. In Table 4, results are obtained after leave-one-out 10-fold cross-validation.

We observe that ConvBERTurk has the highest scores in most of the datasets when original split is used. However, when 10-fold cross-validation is applied, BERTurk has a challenging performance as well. For both setups, XLM-R has the highest performance for the ATISNER dataset; showing that multi-lingual models can be competitive, specifically for spoken queries (utterances).

The performance of Turkish NER on the formal datasets, News Articles and WikiANN-tr, is higher than Tweets, probably due to noisy language. Daily language is not a deteriorating factor in performance, since ATISNER has similar scores as those of News Articles and WikiANN-tr.

The models have slightly better performance in the coarse-grained Tweets dataset, compared to the fine-grained one. This observation is controversy to our expactation of having better scores for coarse-grained.

3.4 Linguistic Analysis

We present a linguistic analysis to investigate the effects of punctuation marks, normalization, lemmatization, and deasciification.

Punctuation Marks Punctuation marks are full stops, apostrophes, question marks, commas, colons, semi-colons, exclamation marks, and quotation marks. We remove all punctuation marks in the corresponding datasets. Our motivation is to observe whether punctuation marks provide any

[5] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.tr.300.bin.gz

[6] https://www.tensorflow.org/

[7] https://simpletransformers.ai/

Models	News Articles			WikiANN-tr			Tweets-CG			Tweets-FG			ATISNER		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
BiLSTM	83.79	76.53	79.12	82.94	86.47	84.66	60.57	60.18	60.14	59.48	62.33	60.47	83.93	87.53	85.59
BiLSTM-CRF	88.67	76.69	80.42	89.22	88.29	88.75	66.31	57.67	61.30	63.68	62.60	62.83	85.77	86.92	86.23
BERT-b-c	83.90	72.84	76.50	90.06	91.16	90.59	64.28	62.86	63.48	64.50	63.85	64.09	89.73	92.99	91.27
RoBERTa-b	85.38	76.35	79.80	89.27	90.18	89.72	60.76	60.67	60.66	62.16	63.39	62.67	90.69	92.96	91.75
XLM-R-b	90.35	83.49	86.39	92.14	92.74	92.44	72.96	76.18	74.44	74.47	77.79	76.00	**91.65**	**94.49**	**92.99**
mBERT-c	87.83	78.86	82.43	92.33	93.27	92.80	68.81	68.93	68.83	70.14	69.90	69.90	90.51	93.67	91.99
distilBERTurk	89.33	83.55	85.88	89.51	90.85	90.17	71.80	73.15	72.43	72.44	74.04	73.15	88.76	92.78	90.67
BERTurk^{32k}	93.49	88.51	90.69	92.25	93.01	92.63	76.74	80.40	78.44	77.47	81.79	79.47	90.00	93.48	91.63
BERTurk^{128k}	92.19	88.12	89.85	90.67	92.98	91.71	**77.09**	81.58	79.20	77.11	82.79	79.77	89.84	93.40	91.54
elecTRa-c	93.86	89.10	91.18	92.36	93.37	92.86	76.03	80.67	78.24	77.02	81.47	79.11	90.05	93.51	91.69
ConvBERTurk	**94.70**	**90.24**	**92.23**	**92.68**	**93.70**	**93.19**	77.05	**82.64**	**79.63**	**78.33**	**83.68**	**80.79**	90.39	93.62	91.92

Table 3: **Comparison of Turkish NER models (original split)**. Models are divided into sub-groups according to neural and tagging models, English language models, multi-lingual models, and Turkish language models. Average of 10 runs on the original split is reported.

Models	News Articles			WikiANN-tr			Tweets-CG			Tweets-FG			ATISNER		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
BiLSTM	87.74	88.79	88.11	83.46	87.60	85.46	57.96	57.29	57.45	57.96	54.77	55.89	88.76	89.34	88.98
BiLSTM-CRF	90.26	89.24	89.59	89.53	89.21	89.36	62.19	58.41	60.01	61.66	58.47	59.70	89.94	90.05	89.92
BERT-b-c	89.00	89.19	88.92	90.51	91.90	91.19	63.55	62.10	62.71	65.19	63.47	64.12	93.41	94.31	93.83
RoBERTa-b	88.15	88.92	88.41	89.74	90.85	90.29	61.66	60.72	61.04	62.53	62.08	62.10	93.73	94.23	93.96
XLM-R-b	92.45	93.07	92.69	92.31	93.18	92.74	72.43	74.46	73.34	73.10	75.39	74.05	94.02	**95.15**	**94.55**
mBERT-c	91.26	91.90	91.48	92.53	93.57	93.04	68.97	67.62	68.19	69.26	69.46	69.18	93.83	94.77	94.28
distilBERTurk	91.10	92.32	91.65	89.91	91.48	90.68	69.86	69.94	69.78	70.46	70.72	70.42	93.02	94.00	93.47
BERTurk^{32k}	93.56	94.50	93.99	92.40	93.44	92.92	75.33	77.58	76.35	75.11	78.92	76.75	93.68	94.63	94.12
BERTurk^{128k}	**94.15**	**94.99**	**94.54**	92.78	93.94	93.35	**76.04**	79.71	77.75	**76.09**	**80.92**	**78.26**	93.58	94.60	94.06
elecTRa-c	93.61	94.77	94.13	92.62	93.72	93.16	74.82	78.13	76.35	74.38	78.80	76.35	93.82	94.50	94.14
ConvBERTurk	93.78	94.95	94.33	**92.97**	**94.16**	**93.55**	75.66	**80.32**	**77.82**	76.06	80.85	78.20	**94.09**	94.96	94.49

Table 4: **Comparison of Turkish NER models (10-fold cross validation)**. Models are divided into sub-groups according to neural and tagging models, English language models, multi-lingual models, and Turkish language models. Average of 10-fold cross-validation is reported.

necessary information for models. An example of removing punctuation marks is to convert "Istanbul'da" (translated to "at Istanbul") to "Istanbulda".

Normalization Normalization is the process for noisy texts to correct wovels and mis-spelling. We use zemberek-python[8] for Turkish normalization. However, we observe that this normalization tool can modify word cases or remove some characters, which might affect the NER performance.

Lemmatization Lemmatization is the process for words to represent them with their dictionary form by grouping inflections. We use zemberek-python[8] for lemmatization process of Turkish datasets. For instance, "oynadılar" (translated to "they played") is converted to "oynamak" (translated to "to play").

Deasciification Deasciification is the process that converts ASCII characters to corresponding Turkish characters. For instance, "c" can be converted to "ç" if necessary. In Turkish, noisy words are mostly written in their corresponding ASCII characters. For instance, instead of "nasılsın?" (translated to "how are you?"), social media users tend to write "nasilsin?". We use turkish-deasciifier[9] for this purpose.

3.4.1 Experimental Setup

In the linguistic analysis, we use the same experimental setup as in Section 3.3.1. Our aim is to compare the results of Turkish NER on raw text with the results after applying punctuation removal, normalization, lemmatization, and deasciification. We employ a neural model, BiLSTM, and Transformer-based model, BERTurk^{32k}, to avoid model-specific results. We report weighted F1 scores on the original splits of WikiANN-tr and Tweets-CG, to compare formal written text and daily noisy language.

[8] https://github.com/Loodos/zemberek-python

[9] https://github.com/emres/turkish-deasciifier

Model	Raw Text		Punctuation		Normalization		Lemmatization		Deasciification	
	Wiki	Tweet	Wiki	Tweet	Wiki	Tweet	Wiki	Tweet	Wiki	Tweet
BiLSTM	84.66	60.14	82.09	**61.76**	78.51	58.14	81.59	53.56	82.61	58.19
BERTurk	92.63	78.44	90.39	**79.50**	90.94	74.24	90.98	73.65	92.41	77.82

Table 5: **Linguistic analysis:** Weighted F1 scores of Turkish NER on raw text, compared to the scores after applying important linguistic steps. Bold scores imply any improvement over raw text.

3.4.2 Experimental Results

The results are given in Table 5. Considering both BiLSTM and BERTurk models, we observe that performance is increased in Tweets dataset when punctuations are removed, but not in WikiANN. The reason could be noisy language in Tweets. Removing punctuation would create more structured text. After applying normalization, lemmatization, and deasciification, the performance is decreased in all cases.

4 Discussion

In this section, we discuss the main insights gained from our experimental results, and then Turkish-specific challenges.

4.1 Main Insights

The main insights can be summarized as follows.

- We report the results on the original train-test splits of all datasets to compare with other studies. In addition, to obtain reliable results, we apply 10-fold cross-validation. We recommend to follow this approach since the performances can change significantly over different splits.
- Transformer language models pretrained with Turkish text data is the current state-of-the-art for Turkish NER. Specifically, ConvBERTurk achieves the highest performance in the majority of cases. We argue that giving attention to spans of text can be more important for NER, compared to self-attention focusing on the whole text.
- Multi-lingual language models have challenging performance in Turkish NER. For instance, XLM-R has the highest performance for ATIS-NER, a spoken query dataset.
- The NER performance in tweets is worse than other domains, possible due to the noisy language used in social media.

4.2 Challenges for Turkish NER

Current datasets that we can access to use have limited size of tokens. Moreover, we find several issues in the datasets that we can access to use for Turkish NER. We apply consistent cleaning and pre-processing steps to all datasets. To understand the capability of generalization of the results to smaller or larger data, there is a need to curate novel and large-scale datasets for low-resource Turkish language. Fine-grained NER datasets are limited; we can only compare the results of fine-grained with coarse-grained in Tweets.

State-of-the-art language models are mostly trained in high-resource languages, such as English. Although there is an effort to train BERT-like models for Turkish (Schweter, 2020), pre-trained language models are still needed for Turkish microblogs, since microblog users can have slang. We similarly observe in our experimental results that the performance is worse in tweets compared to other domains. Alternatively, one can explore translation of Turkish microblogs to high-resource langauges to learn models (Can et al., 2018).

Turkish is a flexible word order language, where one can keep the semantics of context by changing the order of words in a sentence (Oflazer and Saraçlar, 2018). An example is that the meaning of the following sentence "Aziz Sancar'ın Nobel'ini kutluyoruz" (translated as "we celebrate Aziz Sancar's Nobel prize") is the same as its verb is moved to the beginning of the sentence as "kutluyoruz Aziz Sancar'ın Nobel'ini". The models that process text both forward and backward can be a solution for flexible word order, as supported by our experimental results.

5 Conclusion

Named entity recognition is a challenging task that has been widely studied in English. There is a lack of comparative analysis for named entity recognition in Turkish data. In this study, we contribute to the literature in three folds. First, we provide a survey on Turkish NER studies. Second, we compare state-of-the-art NER models on various Turkish datasets that we can access to. Lastly, we analyze a set of linguistic processing steps that would affect the performance of Turkish NER. In future work, we plan to extend our analysis with more datasets and processing steps.

References

Arda Akdemir and Tunga Güngör. 2019. A detailed analysis and improvement of feature-based named entity recognition for turkish. In *International Conference on Speech and Computer*, pages 9–19. Springer.

Emre Kağan Akkaya and Burcu Can. 2021. Transfer learning for turkish named entity recognition on noisy text. *Natural Language Engineering*, 27(1):35–64.

Gizem Aras, Didem Makaroglu, Seniz Demir, and Altan Cakir. 2020. An evaluation of recent neural sequence tagging models in turkish named entity recognition. *arXiv preprint arXiv:2005.07692*.

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Piotr Bojanowski, Édouard Grave, Armand Joulin, and Tomáš Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ethem F Can, Aysu Ezen-Can, and Fazli Can. 2018. Multilingual sentiment analysis: An rnn-based framework for limited data. *arXiv preprint arXiv:1806.04511*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Hakan Demir and Arzucan Özgür. 2014. Improving named entity recognition for morphologically rich languages using word embeddings. In *2014 13th International Conference on Machine Learning and Applications*, pages 117–122. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Beyza Eken and Ahmet Tantuğ. 2015. Recognizing named entities in turkish tweets. volume 5, pages 155–162.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Asim Gunes and A. Cüneyd Tantug. 2018. Turkish named entity recognition with deep learning. In *26th Signal Processing and Communications Applications Conference, SIU 2018, Izmir, Turkey, May 2-5, 2018*, pages 1–4. IEEE.

Onur Güngör, Suzan Üsküdarlı, and Tunga Güngör. 2018. Recurrent neural networks for turkish named entity recognition. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM can solve hard long time lag problems. In *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996*, pages 473–479. MIT Press.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Zihang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving BERT with span-based dynamic convolution. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Michal Konkol and Miloslav Konopík. 2013. Crf-based czech named entity recognizer and consolidation of czech ner research. In *International conference on text, speech and dialogue*, pages 153–160. Springer.

Dilek Küçük and Ralf Steinberger. 2014. Experiments to improve named entity recognition on turkish tweets. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 71–78.

Dilek Küçük and A Yazici. 2009. Rule-based named entity recognition from turkish texts. In *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*, pages 456–460.

Dilek Küçük and Adnan Yazıcı. 2012. A hybrid named entity recognizer for turkish. *Expert Systems with Applications*, 39(3):2733–2742.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 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*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960, Online. Association for Computational Linguistics.

Thomas Mandl and Christa Womser-Hacker. 2005. The effect of named entities on effectiveness in cross-language information retrieval evaluation. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1059–1064.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Chikashi Nobata, Satoshi Sekine, Hitoshi Isahara, and Ralph Grishman. 2002. Summarization system integrated with named entity tagging and ie pattern discovery. In *LREC*.

Kemal Oflazer and Murat Saraçlar. 2018. *Turkish natural language processing*. Springer.

Eda Okur, Hakan Demir, and Arzucan Özgür. 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)*, pages 549–555.

Kezban Dilek Onal and Pinar Karagoz. 2015. Named entity recognition from scratch on social media.

Luiz Augusto Pizzato, Diego Mollá, Cécile Paris, et al. 2006. Pseudo relevance feedback using named entities for question answering.

Lawrence Rabiner and Biinghwang Juang. 1986. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International Conference on Natural Language Processing*, pages 417–427. Springer.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Gökhan Akın Şeker and Gülşen Eryiğit. 2012. Initial explorations on using crfs for turkish named entity recognition. In *Proceedings of COLING 2012*, pages 2459–2474.

Gökhan Akın Şeker and Gülşen Eryiğit. 2017. Extending a crf-based named entity recognition model for turkish well formed text and user generated content 1. *Semantic Web*, 8(5):625–642.

Khaled Shaalan. 2014. A survey of arabic named entity recognition and classification. *Comput. Linguist.*, 40(2):469–510.

Mete Taşpınar, Murat Can Ganiz, and Tankut Acarman. 2017. A feature based simple machine learning approach with word embeddings to named entity recognition on tweets. In *International Conference on Applications of Natural Language to Information Systems*, pages 254–259. Springer.

Serhan Tatar and Ilyas Cicekli. 2011. Automatic rule learning exploiting morphological features for named entity recognition in turkish. *Journal of Information Science*, 37(2):137–151.

Gökhan Tür, Dilek Hakkani-Tür, and Kemal Oflazer. 2003. A statistical information extraction system for turkish. *Natural Language Engineering*, 9(2):181–210.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Reyyan Yeniterzi. 2011. Exploiting morphology in Turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session*, pages 105–110, Portland, OR, USA. Association for Computational Linguistics.

Selim F. Yilmaz, Ismail Balaban, Selim F. Tekin, and Suleyman S. Kozat. 2020. Hybrid framework for named entity recognition in turkish social media. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

Gökhan Çelikkaya, Dilara Torunoğlu, and Gülsen Eryiğit. 2013. Named entity recognition on real data: A preliminary investigation for turkish. In *2013 7th International Conference on Application of Information and Communication Technologies*, pages 1–5.

Furkan Şahinuç, Veysel Yücesoy, and Aykut Koç. 2020. Intent classification and slot filling for turkish dialogue systems. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.