ENHANCING NAMED ENTITY RECOGNITION IN TURKISH BY INTEGRATING EXTERNAL KNOWLEDGE AND EXTRA LAYERS INTO TRANSFORMER-BASED MODELS

by
BUSE ÇARIK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Computer Science

Sabancı University
December 2022

ABSTRACT

ENHANCING NAMED ENTITY RECOGNITION IN TURKISH BY INTEGRATING EXTERNAL KNOWLEDGE AND EXTRA LAYERS INTO TRANSFORMER-BASED MODELS

BUSE ÇARIK

MS in Computer Science THESIS, December 2022

Thesis Supervisor: Asst. Prof. Reyyan Yeniterzi

Keywords: Information Extraction, Knowledge Base, Wikipedia, Twitter

Named Entity Recognition (NER) is a core component in extraction information that aims to detect and classify named entities, such as person and location names. Applications of this task include the detection of named entities in raw texts from various domains. Categorizing news articles, anonymizing texts to ensure privacy, and identifying diseases and drugs from electronic health records in the medical field are some of the usage areas of this task. However, each domain has its own challenges and knowledge requirements. One of the challenging domains in NER is social media because of its noisy nature and context deficiency. In addition, newly named entity classes are included in this domain, covering ambiguous and complex entities such as book or movie titles. Because of these issues, models perform poorly in this domain compared to well-written texts such as news articles.

In this work, we aim to improve the performance of models, particularly in complex entities and lack of context, by integrating external information from a knowledge base, like Wikipedia, into a transformer-based model in an unsupervised manner. To select the external context and add it to the BERT model, we proposed two different methods. In the first approach, the two pipelines called EL_{BERT} and $EL_{MultiBERT}$ attempted to find possible named entities on Wikipedia and utilized the pages they found as external information. Our second method, $EL_{Semantic}$, improved the previous approach by emphasizing the contextually closer pages since detecting every named entity in Wikipedia is not always possible. With EL_{BERT} and $EL_{MultiBERT}$, we achieved significant improvement on the MultiCoNER dataset, which contains

many short samples and complex entities, compared to vanilla transformer-based models. Moreover, by incorporating semantically similar content in the $EL_{Semantic}$, we outperformed the BERTurk model on all datasets with noisy text.

Since the social media datasets in Turkish NER are either old or insufficient, we first constructed a new Twitter dataset. Moreover, since the existing social media datasets have not been evaluated with transformer-based models, we trained variations of these models and compared them with BiLSTM-CRF architecture on social media datasets. We also implemented CRF and BiLSTM layers on top of transformer-based models to improve their performances by capturing relations among labels. The BERT-CRF model outperformed our pipelines with external knowledge, however, it performed poorly compared to our pipelines for the dataset full of short samples and complex entities, namely MultiCoNER. The BERT-BiLSTM-CRF model, on the other hand, performed poorly and lagged behind other transformer-based approaches.

ÖZET

ADLANDIRILMIŞ VARLIK TANIMASINI TÜRKÇE'DE DÖNÜŞTÜRÜCÜ TABANLI MODELLERE HARICI BILGI VE EKSTRA KATMANLARI ENTEGRE EDEREK GELIŞTIRME

BUSE ÇARIK

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS TEZİ, ARALIK 2022

Tez Danışmanı: Asst. Prof. Reyyan Yeniterzi

Anahtar Kelimeler: Bilgi Çıkarma, Bilgi Bankası, Vikipedi, Twitter

Adlandırılmış Varlık Tanıma (AVT), kişi ve konum adları gibi adlandırılmış varlıkları algılamayı ve sınıflandırmayı amaçlayan, bilgi çıkarımının temel görevlerinden birisidir. Bu görevin kullanım alanlarından bazılarına haberlerin kategorize edilmesi, metinlerin gizliliğin sağlanması için anonimleştirilmesi, tıp alanında elektronik sağlık kayıtlarından hastalık ve ilaçların tespit edilmesi örnek olarak verilebilir. Bununla birlikte, her alanın kendine ait zorlukları ve bilgi gereksinimleri vardır. AVT'deki zorlu alanlardan birisi, gürültülü doğası ve bağlam eksikliği nedeniyle sosyal medya verileridir. Ayrıca, kitap veya film başlıkları gibi belirsiz ve karmaşık varlıkları kapsayan yeni adlandırılmış varlık sınıflarının da bu alana dahil edilmesi görevi daha da zorlaştırmıştır. Bu sorunlar nedeniyle modeller, haber makaleleri gibi iyi yazılmış metinlere kıyasla sosyal medya verilerinde daha düşük performans göstermektedirler.

Bu çalışmada, Vikipedi gibi bir bilgi tabanından gelen harici bilgileri denetimsiz bir şekilde dönüştürücü tabanlı bir modele entegre ederek modellerin özellikle karmaşık varlıklarda ve bağlam eksikliğinde performanslarını iyileştirmeyi amaçladık. Dış bağlamı seçmek ve BERT modeline eklemek için iki ayrı yöntem önerdik. İlk yaklaşımımızda, EL_{BERT} ve $EL_{MultiBERT}$ adlı iki yöntemimiz ile Vikipedi'den olası adlandırılmış varlıkları bulmaya çalıştık ve tespit edebildiğimiz sayfalardan harici bilgi olarak yararlandık. Ancak Vikipedi'de adlandırılmış her varlığı tespit etmek her zaman mümkün olmadığı için ikinci yaklaşımımız olan $EL_{Semantic}$'te bağlamsal olarak daha yakın sayfaları vurgulayarak önceki yaklaşımımızı geliştirdik. EL_{BERT} ve $EL_{MultiBERT}$ modellerimiz ile çok sayıda kısa örnek ve karmaşık varlıklar içeren

MultiCoNER veri setinde dönüştürücü tabanlı modellere kıyasla önemli bir gelişme sağladık. Ayrıca, $EL_{Semantic}$ yöntemimizde anlamsal olarak yakın içerikleri eklemeyerek, gürültülü metinlerden oluşan veri setlerinde BERTurk modelinden daha iyi performans elde etmeyi başardık.

Öncelikle Türkçe AVT'deki sosyal medya veri setleri eski ve yetersiz olduğu için yeni bir Twitter veri seti oluşturduk. Dahası, mevcut sosyal medya veri kümeleri daha önce dönüştürücü tabanlı modellerle değerlendirilmediği için bu modellerin varyasyonlarını eğittik ve BiLSTM-CRF mimarisi ile bu veri setleri üzerinde karşılaştırdık. Daha sonra dönüştürücü tabanlı modellerin üzerlerine etiketler arasındaki ilişkileri yakalayarak performanslarını iyileştirmek için CRF ve BiLSTM katmanları uyguladık. BERT-CRF modeli, harici bilgi eklemeyi önerdiğimiz metodlardan daha iyi performans göstermiştir, ancak kısa örnekler ve karmaşık adlandırılmış varlıklarla dolu olan MultiCoNER veri setinde, yöntemimizle karşılaştırıldığında oldukça kötü bir sonuç elde etmiştir. BiLSTM katmanı eklemek ise hiçbir gelişme göstermemiş ve diğer dönüştürücü tabanlı yaklaşımların gerisinde kalmıştır.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor Asst. Reyyan Yeniterzi for her endless patience, guidance, and knowledge on this journey. She has inspired me to become a researcher and motivated me to achieve more. It would not have been possible for me to complete my thesis and master's degree without her continuous support and understanding.

I would also like to thank my best labmate, Fatih Beyhan, for encouraging me to try crazy ideas and for his contributions to this work.

I would like to thank my parents, who were always there for me and supported me even when I was grumpy and stubborn. Lastly, I want to thank my friends, especially Buse Nur Karatepe and Atacan Tütüncüoğlu, for keeping me distracted as much as necessary to protect my mental health.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATONS

BERT Bidirectional Encoder Representations from Transformers . . . 6

BiLSTM Bidirectional Long Short-Term Memory . . . 8, 23

CRF Conditional Random Fields . . . 5, 23

HMM Hidden Markov Model . . . 5

LSTM Long Short-Term Memory . . . 5, 24

NE Named Entities . . . 18, 19, 63, 64

NER Named Entity Recognition . . . 1, 5, 15, 20, 23

NLP Natural Language Processing . . . 6

RNN Recurrent Neural Network . . . 24

1. INTRODUCTION

Named Entity Recognition (NER), one of the steps in extracting information, aims to detect and classify named entities from unstructured texts. Although there is no precise definition of named entities, they can be defined as phrases that refer to a real-world entity, such as proper nouns or numerical expressions (Chinchor, 1998). NER is one of the core tasks that play a crucial role in several challenging natural language processing and information retrieval problems, such as anonymization, entity linking, summarization, and question answering. Therefore, it is a problem that has been studied in Turkish and other languages for a long time, and outstanding results have been achieved with advanced language models in well-written texts such as news articles.

In addition, as online data has increased exponentially in recent years, the demand to extract information from this data has also raised. However, the advanced models cannot achieve exceptional scores on this type of data as in the well-written texts because this domain has different challenges. The first reason that may lead to this performance degradation is the noisy structures of these texts, which can be caused by fast typing and character restrictions. Furthermore, online data frequently lack sufficient context, especially since tweets and search queries are significantly shorter than sentences in news articles. The transformer-based models rely on contextual information, hence, their performance is significantly impacted by the absence of context.

Another difference that causes lower scores in online data is that it expands the label set, consisting of only person, location, and organization names, with new classes that contain ambiguous and fast-growing entities. For instance, newly released movies, TV shows, and songs, which are among the most discussed subjects on online data, are gathered under Product or Creative-Work classes (Malmasi, Fang, Fetahu, Kar & Rokhlenko, 2022a). However, their names can be complicated for models to understand, such as in *Call me by your name*, (i.e., a movie) or *Under the Bridge*, (i.e., a song). In addition to their complexity, they have a rapid growth rate as a new movie or song is released every day. Hence, the majority of these entities cannot be captured in the annotated datasets. As a result, our models have to encounter a large number of examples of these classes that they have never seen before in a real-world setting.

In order to address the lack of context and complex entities, especially in online data, we explored the impact of integrating external knowledge into transformer-based models. Because, in the annotation process, even human annotators leverage online or printed resources when they encounter an unknown phrase. Moreover, several studies utilizing external information with various strategies have improved the performance of transformer-based models significantly in different languages (Wang, Jiang, Bach, Wang, Huang, Huang & Tu, 2021; Yamada, Asai, Shindo, Takeda & Matsumoto, 2020). Hence, to extract relevant information, we suggested new pipelines that find related content from knowledge bases by searching syntactically and semantically similar documents.

Although well-written texts like news articles and Wikipedia pages in Turkish have been extensively researched, studies on noisy texts, such as social media data, are limited. One of the reasons for this is the insufficiency of available datasets in this domain. The existing Twitter datasets in Turkish are either remarkably small (Küçük & Can, 2019) or created a long time ago (Çelikkaya et al., 2013). Due to Twitter's restrictions on data sharing, tweets are shared with their unique ID rather than their text. As a result, if users delete tweets that are part of the annotated data, it is not possible to access those tweets again. As the deletion rate of tweets and users increases over time, a significant amount of data is lost in old datasets. Therefore, as a part of this study, we built a new Twitter dataset consisting of recent tweets. Furthermore, to prevent the dataset from being discarded in the future, we shared the models trained with these tweets publicly.

Moreover, transformer-based models have not been explored for social media datasets in detail, as well-written texts. Hence, we experimented with several transformer-based models on noisy texts, including our new Twitter dataset, and compared our results with BiLSTM-CRF architecture. Thus, we could compare the performances of these models on well-written and noisy texts in the same experimental setting. Furthermore, to improve the performance of transformer-based models, we implemented additional layers, like CRF and BiLSTM, to these architectures.

1.1 Research Questions

In this thesis, in order to improve the current state-of-the-art models for NER in Turkish, especially in noisy text, we focus on integrating external knowledge from a knowledge base like Wikipedia and adding extra layers on top of these models. While studying these approaches, we aimed to give answers to the following research questions.

RQ1: Would implementing extra layers on top of the transformer-based models outperform only using the softmax function?

RQ2: Would integrating an external context from a knowledge base into transformer-based models improve results for NER in Turkish, especially for noisy texts?

We ask the following follow-up questions to measure the effectiveness of using external knowledge in the NER task.

- **RQ2-1:** In what conditions would utilizing the Wikipedia pages of detected possible named entities as external knowledge improve the NER?
- **RQ2-2:** Would highlighting semantically similar texts to input samples be more effective than using only pages of detected possible named entities?

Our experiments on six datasets showed that including external knowledge improved the BERT's performances, especially in social media and search query domains. We also compared the effectiveness of our approach with BiLSTM-CRF, BERT-CRF, and BERT-BiLSTM-CRF models. Our contributions are summarized as follows:

- We introduced a new Turkish NER dataset enriched with new named entity classes in the social media domain.
- We also shared the models that we trained with this dataset publicly on HuggingFace[1].
- Transformer-based models, which hold state-of-the-art results in several datasets in Turkish, were evaluated, including unattempted social media datasets. Moreover, the effect of CRF and BiLSTM layers on top of transformer-based models was compared in the same environmental setting.
- We proposed two new methods to integrate external knowledge into transformer-based models. First, we introduced two pipelines, called EL_{BERT} and $\text{EL}_{\text{MultiBERT}}$, that provide Wikipedia pages of possible named entities as external context. Second, we utilized semantically similar content as additional

[1] https://huggingface.co/

information in the $\text{EL}_{\text{Semantic}}$ pipeline.

1.2 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 provides background information about the NER task and describes work done by the previous studies for NER in Turkish and other languages. In chapter 3, the datasets used in this thesis are explained in detail. In chapter 4, the neural models that we used in this study are explained, and their results are reported. Chapter 5 describes our methodologies that utilize external knowledge and presents the results of these methods. In chapter 6, we discuss and analyze the performances of models described in chapters 4 and 5. Finally, chapter 7 concludes this thesis by summarizing the research findings and exploring possible future directions.

2. BACKGROUND AND RELATED WORK

In this chapter, we give an overview of approaches to address that named entity recognition task throughout the years. Then, we discuss the challenges of social media data, in general, noisy texts. Furthermore, studies that utilized knowledge bases and gazetteers have been introduced. Finally, we present studies on Turkish NER in various domains.

Early approaches in NER were based on handcrafted features, such as case information, part-of-speech tags, and rule-based approaches (Humphreys, Gaizauskas, Azzam, Huyck, Mitchell, Cunningham & Wilks, 1998; Krupka & IsoQuest, 2005). Rules generated based on patterns, domain information, and linguistic characteristics are combined with gazetteers or lexicons (Li, Sun, Han & Li, 2020). However, because the rules are domain or language-specific, they are difficult to fit across different datasets.

To address this problem, statistical and supervised machine learning approaches turned NER into a sequence labeling task. Rules in these models are created automatically using features extracted from the large annotated data. The popular systems used for this task are as follows: Hidden Markov Model (HMM) (Bikel, Miller, Schwartz & Weischedel, 1997; Eddy, 1996), Support Vector Machines (SVMs) (McNamee & Mayfield, 2002; Takeuchi & Collier, 2002), Maximum Entropy Markov Models (McCallum, Freitag & Pereira, 2000), and Conditional Random Fields (CRF) (Finkel & Manning, 2009; Krishnan & Manning, 2006; Lafferty, McCallum & Pereira, 2001; McCallum & Li, 2003). The CRF model has also been experimented with in the social media domain (Liu, Zhang, Wei & Zhou, 2011; Ritter, Clark, Mausam & Etzioni, 2011).

Constructing features manually from raw data is one of the issues with data-driven techniques. Neural network models have replaced these techniques by eliminating the feature engineering part. While different architectures like convolution neural networks (CNN) were employed on NER (Collobert & Weston, 2008), bidirectional LSTM models on top of the CRF model (BiLSTM-CRF) achieved an outstanding performance (Collobert, Weston, Bottou, Karlen, Kavukcuoglu & Kuksa, 2011; Huang, Xu & Yu, 2015). In order to learn the word representations, unsupervised continuous skip-gram and bag-of-words (CBOW) models trained on large data collections have been used (Mikolov, Chen, Corrado & Dean, 2013; Pennington, Socher & Manning, 2014). To extract character-level features, CNN architecture has applied, then constructed character embeddings are fed into the BiLSTM-CRF model with word embeddings (Chiu & Nichols, 2016). The effectiveness of BiLSTM-CRF and character embeddings have also been experimented with in different languages (Lample, Ballesteros, Subramanian, Kawakami & Dyer, 2016; Zhang & Yang, 2018). The drawback of these architectures is that the same tokens with different meanings are represented with the same embedding vectors. To include the context information, (Akbik, Blythe & Vollgraf, 2018) implemented a character language model that processes each character forward and backward with two LSTM layers.

Transformer-based models have become the new standard of NLP due to their outstanding performance on various problems. There are several advantages of these models. The famous one, BERT, learns contextual representation by predicting the randomly masked tokens on an immense corpus. It can also be applied to different tasks by only changing the last layers. Weights of the models trained with multilingual corpus, namely mBERT, and modified versions of BERT with a larger multilingual data collection, namely XLM-R (Alexis, Kartikay, Naman, Vishrav, Guillaume, Francisco, Edouard, Myle, Luke & Veselin, 2019), are publicly available. Furthermore, using the CRF model on top of BERT to predict label sequence improved results in different languages (Arkhipov, Trofimova, Kuratov & Sorokin, 2019; Souza, Nogueira & Lotufo, 2019).

2.1 Named Entity Recognition for Noisy and Short Texts

The recent advanced neural models achieved outstanding results on well-written texts, especially in the news articles domain; however, their performance degrades drastically when applied to short, noisy texts such as social media texts and search queries (Meng, Fang, Rokhlenko & Malmasi, 2021). One of the reasons for this decrease in the social media domain is using special symbols, such as mentions, emojis, and hashtags. Moreover, there are a lot of misspellings, grammar mistakes, and arbitrary abbreviations due to length limit or typing fast. The capitalization rule, which is an important indicator for the NER task, is mostly ignored in these kinds of texts. For instance, a study (Mayhew, Tsygankova & Roth, 2019) demonstrated that hiding capitalization information in training caused the F1 score to drop from 92.54% to 34.46% in the CoNLL test set containing case information.

In addition to the writing style, there are other factors explaining the performance gap between these two domains. First, state-of-the-art transformer-based models like BERT heavily depend on contextual information; on the other hand, the context in social media texts can be missing due to their short nature (Meng et al., 2021). Furthermore, while the named entity types in well-written texts like news articles are well-known and frequently encountered entities, namely person, location, and organization, in the social media domain, complex and ambiguous entities, such as movie and song names, are included (Fetahu, Fang, Rokhlenko & Malmasi, 2022; Meng et al., 2021). Moreover, although the number of named entities in the training set of news articles is high, they are often repetitions of the same entity, and the number of unseen entities in their test set is low, which led to outstanding performances in the news domain, as stated in Augenstein, Derczynski & Bontcheva (2017). However, while the total number of named entities in noisy texts is low, their test set has a large number of unseen named entities.

2.2 Named Entity Recognition with External Knowledge

> [Washington | PER] was the first president of the United States. It was decided that the last meeting would be held in [Washington | LOC].
>
> [Fenerbahçe | ORG] played well in the soccer game with [Beşiktaş | ORG].

As stated in Ratinov & Roth (2009), NER is a task that requires knowledge as well as context. For example, we need context to understand whether Washington is referred to as a person or a place in the first sentence above. However, in the second sentence, although there is a context, we cannot decide whether Fenerbahçe is a soccer team or a player without external knowledge. It is particularly difficult to identify named entities from other languages, such as foreign organizations or locations, as in this example. Furthermore, even annotators might be unsure about a phrase in the annotation process. To learn the meaning of unknown words, they utilize additional resources like Google (Wang et al., 2021). Therefore, studies have attempted to integrate external knowledge like gazetteers and lexicons into their methodologies. Including such knowledge has improved both statistical and neural approaches for this task (Yamada et al., 2020).

Early approaches utilized external knowledge by giving it as a separate feature or encoding it in embedding vectors. Ratinov & Roth (2009) improved the scores on the CoNLL-2003 news dataset by adding a discrete feature based on tokens' presence in gazetteers that were constructed from Wikipedia pages. Passos, Kumar & McCallum (2014) utilized lexicon information at the word embedding level by extending the skip-gram model. Their updated word embedding model not only predicts the next token but also determines if a phrase is in the lexicon. Another study encoded the information of whether a token is a named entity or part of the named entity into the feature vectors alongside character features learned from CNN architecture (Chiu & Nichols, 2016). Their stack BiLSTM network trained with character and word embeddings enhanced with lexicon information achieved new state-of-the-art results in the CoNLL-2003 and OntoNotes 5.0 news article datasets.

Liu, Yao & Lin (2019) also utilized external knowledge in neural architectures, but instead of including it directly at the embedding level, they created a separate module. A hybrid semi-Markov Conditional Random Fields model was enhanced by a sub-tagger module that trained as a span classifier. This new module was used as a soft dictionary lookup that returned a score for each token according to whether they matched with an item in the gazetteer.

Ding, Xie, Zhang, Lu, Li & Si (2019) introduced a multi-digraph structure that captures the relation between characters and gazetteers to improve the matching accuracy between entities and gazetteers in Chinese. The constructed digraph is given to a Gated Graph Neural Sequence to solve the ambiguities in the matches and LSTM-CRF structure for final predictions.

Lin, Lu, Han, Sun, Dong & Jiang (2019) created an auxiliary task called gazetteer network that aims to detect named entities from given phrases. The representation of a phrase learned by a gazetteer network was then included in an attentive neural network (ANN) that utilized the attention mechanism. ANN model enhanced by the gazetteer network outperformed the single ANN in a news-related dataset.

Recent studies have applied external knowledge to transformer-based models to improve their performances. (Song, Lawrie, Finin & Mayfield, 2020) created one-hot vectors for each word in their training set that matched with items in their gazetteer produced from Wikidata. The one-hot vectors generated for each named entity type are concatenated with the last hidden layers of frozen BERT and fed into the BiLSTM-CRF structure. BERT-BiLSTM-CRF achieved better results when gazetteer features were added for English and Chinese news articles. However, the system is constrained by the fact that one-hot vectors cannot hold information about context and spans.

Besides integrating external knowledge in downstream tasks, other recent studies utilized knowledge bases directly in the pretraining process of transformers. KnowBERT (Peters, Neumann, Logan, Schwartz, Joshi, Singh & Smith, 2019) is an enhanced version of the BERT model that inserts entity representations between the BERT's layers. LUKE (Yamada et al., 2020) is an adaptation of the RoBERTa model that aims to learn contextualized entity representations. In this study, a new pretraining task was introduced in which it predicts the masked entities of given sentences along with masked tokens on a large corpus generated from Wikipedia. Although these modified transformer-based models outperformed in various information extraction tasks, including NER, they require retraining, which costs enormous computational resources, with each update to the knowledge bases.

Previous studies have also leveraged gazetteers and knowledge bases as external information to address the challenges of social media, such as complex entities and lack of context. An early study (Yamada, Takeda & Takefuji, 2015), adapted the entity linking task to solve NER in a Twitter dataset. The system aims to identify possible entities by searching Wikipedia entries with inputs in the form of n-grams to provide external knowledge. To cope with the noisy structure of tweets, several string-matching algorithms like a fuzzy match were implemented. It utilized a random forest algorithm both for matching the candidate mention with entries in the knowledge base and assigning named entity labels to the possible entities.

Manchanda, Fersini & Palmonari (2015) also improved NER performance in social media with an end-to-end named entity linking pipeline. Their NER system was boosted by the information acquired from knowledge bases. The predictions of the NER model were re-classified with the information obtained after the relevant texts were retrieved by searching the knowledge bases.

A recent study (Wang et al., 2021) also approached this task as an entity linking problem for various domains, including news, social media, and biomedical. The relevant texts retrieved from a knowledge base using input sentences were sorted based on their contextual similarity to use as external knowledge. The most contextually similar pages are concatenated with the input sentence and fed into a transformer-based model. The output representation obtained from a transformer-based model is given to a CRF layer to predict the output sequence.

Another research on noisy text (Fetahu, Fang, Rokhlenko & Malmasi, 2021) proposed a system that aims to add context to information from gazetteers, which was missing in previous studies. While their architecture learns the word representation from BERT, a gazetteer module consists of BiLSTM layers that acquire context from the matched entities. They presented a gated architecture that decides which one to trust more to combine information from these two modules. Furthermore, Meng et al. (2021) adapted the system proposed by Fetahu et al. (2021) for web queries that are contextually short and contain entities from other languages. To address these code-mixed terms, multilingual gazetteers were given to the gated architecture, and the word representation model was replaced with XLM-R, which can compete with monolingual models. Later, Fetahu et al. (2022) also adopted and evaluated the architecture of (Fetahu et al., 2021) for cross-lingual and cross-domain scenarios.

2.2.1 SemEval Shared Task on Multilingual and Complex NER

A shared task in 2022 was undertaken to evaluate recent architectures on detecting named entities, particularly complex and ambiguous ones, such as movie or product names in a low-context and multilingual environment (Malmasi, Fang, Fetahu, Kar & Rokhlenko, 2022b). The data used in this task include sentences from Wikipedia, search queries, and questions in eleven languages. The main challenge in this task was identifying the complex entities which were included in the datasets with Creative Work, Group, and Product entity types. Since these types can be semantically ambiguous and their new instances increase rapidly, it is hard to identify with classical approaches. Therefore, the test set was constructed to be significantly larger than the training set in order to demonstrate the generalizability of NER models on unobserved and complicated entities. Another challenge with the test set was that the majority of the extremely short samples caused participants to encounter a lack of context.

The task, in which 55 teams participated, achieved the top performance when transformer models were enhanced with external knowledge. Creative Work and Group entity types benefited the most from external knowledge. In addition to the external context, the models were improved by using the ensemble technique.

Winner in multilingual track and almost all languages (Wang, Shen, Cai, Wang, Wang, Xie, Huang, Lu, Zhuang, Tu, Lu & Jiang, 2022) retrieved relevant pages from Wikipedia by giving an input sentence as a query. The content of these pages was used as additional context by concatenating with input sentences. The concatenated inputs were fed into the XLM-R model, followed by a CRF layer to produce named entity classes. The final predictions were decided with majority voting among multiple XLM-R - CRF models with different seeds.

Chen, Ma, Qi, Guo, Ling & Liu (2022), which ranked second in most of the tracks, integrated a gazetteer network into a pre-trained model. Unlike previous studies, they aimed that the gazetteer network learns the semantic representation of the entities rather than simply using it as a presence indicator. The architecture was trained in two stages. In the first stage, the gazetteer network consists of a dense and BiLSTM layer fed with one-hot vectors constructed for each token based on their presence in the gazetteer. Meanwhile, the frozen XLM-R produced word embeddings for each token in the input sentence. The first stage aims to teach the gazetteer network from the embeddings of the pre-trained model by reducing the KL divergence loss between the outputs of these two networks. In the second stage, these two networks were trained together to obtain the final predictions with a classifier head fed by the combination of outputs of these two networks.

Ma, Jian & Li (2022), which ranked third on the English track, concatenate entities matched from LUKE's entity dictionary and their corresponding types with the input sentence to learn a contextual representation of entities by a BERT model. They also introduced an auxiliary task that decides whether a token is a named entity. Moreover, to improve the performance in named entity classes that is hard to distinguish, like Creative Work and Product, KL divergence loss among the examples of these types over logit matrix was included in the loss function.

Our approach to addressing this task for the Turkish track is utilizing Wikipedia as an external context to enhance the BERT's performance pre-trained on Turkish corpora (Çarık, Beyhan & Yeniterzi, 2022). We achieved third place in this track with our $\mathrm{EL}_{\mathrm{MultiBERT}}$ approach.

2.3 Named Entity Recognition in Turkish

The agglutinative structure of Turkish poses its challenges as in the other morphologically rich languages such as Finnish or Czech. First, a countless number of different words can be derived from a root by adding morphemes and suffixes. Due to the derivational nature of Turkish, the vocabulary size increases remarkably, which may cause the models to encounter many unknown words. For example, the number of unique words in a large Turkish dataset (Tür et al., 2003) with more than 10 million tokens is 474,957; however, this number drops to 97,734 unique words after stemming from each token (Aras, Makaroğlu, Demir & Cakir, 2021). The fact that derivative words might have multiple meanings is another issue. The problem of the data sparsity and ambiguity of meaning led researchers to utilize morphological analysis to NER in Turkish (Oflazer, Göçmen & Bozşahin, 1994).

While earlier studies on NER in Turkish focused on rule-based and statistical approaches, with the rise of the deep learning era, it shifted to neural and large transformer-based models. We can examine the previous works in three categories as traditional, deep learning, and transformer-based approaches.

2.3.1 Traditional Approaches

Early studies in Turkish NER employed rule-based approaches and statistical models like HMM (Rabiner & Juang, 1986) and CRF (Lafferty et al., 2001). Tür et al. (2003) is the first study that both experiment with HMM model and create the first dataset on Turkish NER using news articles. This study was followed by several rule-based methods. Küçük & others (2009) and Küçük & Yazici (2009) generated pattern-based rules and lexical resources from news articles and applied their method in other domains, such as children's books, historical texts, and financial texts. Another developed a system that automates rule generation with a supervised learning technique (Tatar & Cicekli, 2011). Later, Küçük & Yazıcı (2012) improved its rule-based system with a hybrid approach combining rule-based and rote learning that remembers the named entities encountered in the training data. Meanwhile, other studies explored the CRF model utilizing the agglutinative structure of Turkish. Yeniterzi (2011) experimented with the CRF model by tokenizing words at the morpheme level and feeding morphological features as separate tokens. Another study (Şeker & Eryiğit, 2012; Seker & Eryigit, 2017) boosted the CRF model with a considerable amount of lexical and hand-crafted morphological features besides large gazetteers.

Furthermore, the CRF model enriched with morphological features and gazetteers proposed in (Şeker & Eryiğit, 2012) applied to informal texts adding a text normalization step (Çelikkaya et al., 2013; Seker & Eryigit, 2017). Şeker & Eryiğit (2012) is also the first study that introduced new datasets on noisy texts. Also, the rule-based approaches (Küçük, Jacquet & Steinberger, 2014; Küçük & Steinberger, 2014) were implemented for tweets by modifying capitalization and diacritic constraints. Eken & Tantuğ (2015) introduced a larger Twitter dataset and employed a CRF method similar to the (Şeker & Eryiğit, 2012). However, instead of morphological features, they gave suffixes to the model as separate tokens, and features from gazetteers were extracted with a distance-based matching algorithm.

2.3.2 Deep Learning Approaches

The requirement of the manual feature engineering and language-dependent strategies of traditional approaches led researchers to the deep learning methods. Unsupervised neural networks like Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski, Grave, Joulin & Mikolov, 2017) were used to learn the representation of words and their relations between neighbors. While Demir & Özgür (2014) developed a regularized averaged perceptron (Ratinov & Roth, 2009) that utilizes word embeddings and language-independent features like capitalization, Okur, Demir & Özgür (2016) applied a similar approach to the informal text. Another study (Onal & Karagoz, 2015) generated word embeddings from scratch with a mixed corpus to feed into a neural network. In later studies, different levels of word embeddings, such as words, characters, or morphological features, were investigated with CRF on top of BiLSTM architecture to capture the long-term dependencies (Güneş & Tantuğ, 2018; Güngör, Üsküdarlı & Güngör, 2018; Kuru, Can & Yuret, 2016). To increase performance on noisy text, Akkaya & Can (2021) applied transfer learning through a CRF layer trained on a larger formal text. A recent study (Çetindağ, Yazıcıoğlu & Koç, 2022) introduced the first dataset on legal documents and experimented with different embedding vectors, including Morph2Vec, GloVe, and character embeddings using BiLSTM-CRF.

2.3.3 Transformer-based Approaches

Recently the Turkish NER community has become interested in transformer-based models like BERT because of their outstanding performance across a range of tasks and languages. Recent studies have experimented with the multilingual and Turkish BERT models and a CRF layer on top (Aras et al., 2021; Ozcelik & Toraman, 2022; Safaya, Kurtuluş, Goktogan & Yuret, 2022). As Aras et al. (2021) and Safaya et al. (2022) applied these architectures to well-written text like news articles and Wikipedia pages, Ozcelik & Toraman (2022) investigated variations of these models on both well-written and noisy text with detailed error analysis.

3. DATASET

In this chapter, we describe the data collection and annotation process of our new Twitter dataset, namely TW-SUNLP. We also provide a detailed explanation of the datasets from various domains and writing styles in the Turkish NER literature utilized in this study. To use in our experiments, we acquired five datasets from the following studies: Milliyet (Tür et al., 2003), WikiANN (Pan et al., 2017), TW-2013 (Çelikkaya et al., 2013), IWT (Seker & Eryigit, 2017), and MultiCoNER (Malmasi et al., 2022a). The remaining datasets were not included in this study as we cannot access them (Eken & Tantuğ, 2015; Küçük et al., 2014; Tatar & Cicekli, 2011).

3.1 TW-SUNLP

The publicly available Twitter datasets in Turkish NER are limited. The available ones are either too small or old (Küçük & Can, 2019; Küçük, Küçük & Arıcı, 2016). Besides the fact that old datasets contain outdated topics, there is another problem that hinders their use. Since Twitter does not allow sharing tweets, researchers can only release the IDs of the tweets. Hence, deleting tweets or accounts causes most of the annotated data to be lost over time. In recent datasets, however, the number of instances is significantly small (e.g., around 1,8K in Küçük & Can (2019)) to train a model. Furthermore, the named entity types in the current studies are limited to PLO, TIMEX (i.e., Date and Time), and NUMEX (i.e., Money and Percentage) classes. However, there are various complex named entities such as movies, song titles, or products on social media. Another problem with the present datasets is that most of them were gathered in a short period of time. Therefore, the diversity of named entities in these datasets is limited, as the number of entities in tweets is also generally low. On the other hand, the demand for detecting named entities in tweets has increased as a result of the growth of social media content.

To address these problems, we created a new Twitter dataset called TW-SUNLP. In order to build a diverse dataset in terms of both topics and named entities, we applied the following steps in the data collection and filtering processes:

- Collected 65 million tweets by filtering the popular topics discussed on Turkey's Twitter between June 2020 and June 2021.

- Removed the duplicate tweets without taking into consideration of mentions, hashtags, and URLs.

- Selected tweets with more than 50 characters long to improve the possibility of a named entity.

- Fine-tuned the pre-trained BERTurk model with the largest NER dataset in Turkish, namely Milliyet. Named entity class predictions were acquired from this model for each tweet in our collection and selected tweets that have at least one unseen named entity among the collection based on these predictions.

- Set a limit that any hashtag can be in a maximum of 3 tweets to ensure that we have a dataset with diverse topics.

- Selected 5,000 tweets randomly for annotation from the remaining tweets after the above steps.

As a result of these methods, TW-SUNLP became the largest Twitter dataset in terms of word and named entity counts. Moreover, due to the minimum character restriction, the average sample length of this dataset is 25.2, which is significantly higher than the other Twitter dataset. In addition, thanks to these steps, the number of unique entities in our dataset is 10 times higher than in the other Twitter dataset.

In the annotation process, we included new named entity types, namely Product and TV-Show, in addition to Person, Organization, Location, Money, and Time classes. Items created by individuals or businesses are included in the Product class. Examples of this category are movies, novels, and Facebook. Soap operas, reality shows, and other TV series that were shown on TV are categorized under the TV-Show class since they are discussed often on Twitter. The examples of these entity classes from our dataset are shown below.

- *Aklıma Kurtlar Vadisindeki*$_{\textbf{TV-SHOW}}$ *Laz Ziya*$_{\textbf{PERSON}}$ *geldi #OyAsiyemOy*

- *Yıkılsın Tweeter*$_{\textbf{PRODUCT}}$ *bu gece Enis Talha*$_{\textbf{PERSON}}$ *yoğun bakımda #KulüplereDegilEniseNefesOl*

We have an annotation team of four undergraduate students. Each tweet was annotated by two students. At the end of the annotation process, we achieved a high agreement score with a 0.87 Cohen Kappa score. Additionally, we published data collection and annotation process and baseline scores of this dataset at the Language Resources and Evaluation Conference[1] (Çarık & Yeniterzi, 2022). Also, we shared the weights of transformer-based models trained with this dataset publicly in the HuggingFace library[2], as tweets can be lost over time.

3.2 Available Datasets

The available datasets we experimented with in this study can be separated into two categories as formal and informal. Formal texts adhere to grammar and syntactic rules, such as capitalization. Examples in this category include Wikipedia pages and news articles. Noisy texts like social media posts (e.g., tweets) filled with misspellings and code-mixed phrases are classified as informal.

Dataset statistics are reported in Table 3.1 and Table 3.2, which are divided as formal and informal. Data distributions, including validation sets, are reported in Appendix 7.1. All formal datasets were annotated with Person, Location, and Organization (PLO) named entity types, gathered under the name of ENAMEX format by the Message Understanding Conference (MUC) series (Grishman & Sundheim, 1996). Named entities in informal datasets have been extended to include more complex and ambiguous entities, such as Product and TV-Show, as well as numerical and temporal expressions like Time and Money. Details about each dataset are explained in the following sections.

3.2.1 Formal Datasets

Milliyet (Tür et al., 2003) is the oldest but the largest manually annotated dataset in Turkish NER. The dataset consists of news articles collected between 1997 and 1998, with more than 27 thousand sentences. It is the largest dataset in terms of

[1] https://lrec2022.lrec-conf.org/en/

[2] https://huggingface.co/busecarik/berturk-sunlp-ner-turkish and https://huggingface.co/busecarik/bert-loodos-sunlp-ner-turkish

token size, with more than 500 thousand words. It was labeled manually with PLO classes as one of the formal datasets. Since Safaya et al. (2022) provided their splits for this dataset publicly in their repository[3], we used the same split in this study to make our results comparable.

WikiANN (Pan et al., 2017), also known as PAN-X, is a cross-lingual dataset that contains 282 languages. It consists of Wikipedia pages that are labeled semi-automatically. Again for comparability, we retrieved the WikiANN from the same repository as Milliyet. The dataset labeled in ENAMEX format consists of 20,000 sentences for training and 10,000 for both validation and testing. Although the Milliyet is the largest dataset in terms of token count, the semi-automatically annotated WikiANN contains substantially more named entities.

Table 3.1 Statistics on the formal datasets used in this study.

	Milliyet		**WikiANN**	
	Train	Test	Train	Test
Person	14,690	1,603	13,207	4,519
Location	9,763	1,126	14,693	4,914
Organization	9,158	873	12,099	4,154
Sentences	24,820	2,751	30,000	10,000
Tokens	465,528	49,600	225,716	75,731
NE	33,611	3,602	39,999	13,587
Unique NE	9,148	1,466	24,998	9,045

3.2.2 Informal Datasets

We have four informal datasets, including ours, from different domains such as social media, online blogs, and search queries. The named entity classes vary between these datasets.

TW-2013 (Çelikkaya et al., 2013) has 5039 tweets that were annotated with TIMEX, and NUMEX, in addition to ENAMEX. The dataset was re-annotated by Seker & Eryigit (2017) to enhance its quality, and we used the updated version of this dataset in our study. There are two modifications that we applied to this dataset. First, we did not consider mentions (i.e., @...) labeled as Person because it is ambiguous

[3] https://data.tdd.ai/#

Table 3.2 Statistics on the informal datasets, including our Twitter dataset, in this study.

	TW-2013		**IWT**		**TW-SUNLP**		**MultiCoNER**	
	Train	Test	Train	Test	Train	Test	Train	Test
Person	622	75	357	23	4,752	830	4,645	26,876
Location	165	19	236	24	1,031	194	6,155	34,609
Organization	390	34	369	32	2,520	457	-	-
Money	8	4	43	2	153	17	-	-
Date	49	7	52	7	-	-	-	-
Time	18	2	7	2	528	105	-	-
Percent	2	1	7	1	-	-	-	-
TV-Show	-	-	-	-	256	48	-	-
Product	-	-	-	-	335	45	3,342	21,388
Group	-	-	-	-	-	-	3,735	21,951
Corporation	-	-	-	-	-	-	2,909	21,137
Creative Work	-	-	-	-	-	-	3,764	23,408
Sentences	4,535	504	4,509	502	4,250	750	16,100	136,935
Tokens	42,026	4,722	42,692	4,667	106,440	18,702	229,816	723,226
NE	1,254	142	1,071	91	9,575	1,696	24,550	149,369
Unique NE	905	120	646	72	6,252	1,278	12,654	87,701

whether a person or an organization and their use violate the protection of personal information. Secondly, we consider only the named entities that have the highest span if there are nested named entities. We randomly divided 10% for testing, as there was no version previously split the data as training and test. When we compare this Twitter dataset with ours in Table 3.2, we observe that their differences are not limited to the average sentence length. Due to the steps followed in the data collection process, the number of total and unique named entities in our Twitter dataset is significantly higher than the TW-2013.

IWT (Seker & Eryigit, 2017) consists of user-generated content from several domains, including customer reviews, social media posts, blogs, and forums collected from ITU Web Treebank. The label set of this dataset is the same as TW-2013, that is, it consists of ENAMEX, TIMEX, and NUMEX. Also, we again randomly selected %10 of the data for testing. Although this dataset is classified under the informal datasets since it was collected from online data, it has fewer writing errors than Twitter datasets.

MultiCoNER (Malmasi et al., 2022a) is an automatically annotated multilingual dataset containing Wikipedia pages, search queries, and questions, constructed to emphasize the following challenges of NER: i) lack of context in short texts, ii) linguistically complex entities like movie and song names, iii) training datasets do not reflect the real-world diversity of named entities classes, iv) texts contain terms or phrases from other languages. In order to address these problems, they built a very large test set filled with significantly short samples. For instance, the average sentence length in both training and validation is 14.27, but this number drops to 5.28 in the test set. Comparing the number of sentences and tokens in different sets also illustrates this issue. While the test set has more than 130,000 samples, the training set only contains 15,300. However, while there are more than 218K tokens in the training set, the test set has only 723,226 words despite a large number of instances. In addition, to emphasize problems with foreign words, several code-mixed terms were included in both training and test sets. Besides, its label set is enhanced to capture the complex and ambiguous named entities such as movies or songs named with Creative Work and Product types. Since this dataset has the characteristics of an informal dataset, even though it contains Wikipedia pages, we categorized it as informal.

One of the important differences between formal and informal datasets that we can observe in Tables 3.1 and 3.2 is the difference between the total number of named entities and the number of unique entities. Although the total number of named entities in the formal datasets is substantially higher than the informal ones, when comparing the ratio between the total and unique named entities, the entities in informal ones are more diverse. The highest gap between these two is in the news articles domain, which obtains the highest score with more than 95%. This observation shows the effect of memorization on performance increase in this domain (Augenstein et al., 2017).

3.3 Data Preparation

Since the named entities consist of one or more words, we adopted **IOB2**, also known as **BIO**, annotation scheme built for the NER task to demonstrate the span of the named entities (Sang & Veenstra, 1999). The label of the first token of the named entity starts with **B-**, represents the entity's beginning, and is followed by its named entity class. If the entity contains more than one word, the subsequent tokens start with **I-** instead of **B-**. And it is also followed by the type of its entity class. An example of this annotation scheme is shown in Table 3.3.

Table 3.3 Example of the IOB2 format.

Tokens	IOB2 tags
Ali	B-PERSON
Koç	I-PERSON
başvuruda	O
bulunduğu	O
Türkiye	B-ORGANIZATION
Futbol	I-ORGANIZATION
Federasyon	I-ORGANIZATION
’	O
undan	O
haber	O
bekliyor	O

3.4 Evaluation

As evaluation metrics, precision, recall, and averaged F1 scores according to standard CoNLL[4] were implemented with seqeval (Nakayama, 2018) library. The evaluation metrics were calculated at the entity level, which means that the model has to correctly identify all tokens within the span of an entity. For example, for the sentence *'Dolmabahçe Palace is located in Istanbul'*, Dolmabahçe Palace is one of the named entities in this sample. In order to produce a correct prediction for this named entity, the model must label Dolmabahçe as *B-LOCATION* and Palace as *I-LOCATION*. If one of the tokens of the entity is labeled wrong, there is no partial point, even if the other tokens are predicted correctly.

3.4.1 Results from the Previous Literature

We illustrated the results of previous studies on datasets that we used in this study in Tables 3.4 and 3.5, separated as formal and informal. The results for Milliyet-NER are not fully comparable, as the train and test separations were different in previous studies.

[4] The Conference on Natural Language Learning that is organized by SIGNLL (ACL’s Special Interest Group on Natural Language Learning)

Table 3.4 F1 scores of previous studies on datasets of news articles and Wikipedia pages, named Milliyet-NER (Tür et al., 2003) and WikiANN (Pan et al., 2017), respectively.

	Method	Milliyet-NER	WikiANN
Tür et al. (2003)	HMM	91.56	-
Yeniterzi (2011)	CRF	88.94	-
Şeker & Eryiğit (2012)	CRF	91.94	-
Demir & Özgür (2014)	Reg. Avg. Perc.	91.85	-
Kuru et al. (2016)	BiLSTM-CRF	91.30	-
Güngör et al. (2018)	BiLSTM-CRF	93.37	-
Güneş & Tantuğ (2018)	BiLSTM-CRF	93.69	-
Aras et al. (2021)	BERTurk-CRF	95.95	-
Safaya et al. (2022)	BERTurk-CRF	96.48	93.07
Ozcelik & Toraman (2022)	ELECTRA-tr	96.10	91.91
Ozcelik & Toraman (2022)	ConvBERTurk	95.88	92.26

Table 3.5 F1 scores of previous studies on informal text, named TW-2013 (Çelikkaya et al., 2013) and IWT (Seker & Eryigit, 2017). *TW-2013 was re-annotated by Seker & Eryigit (2017). In (Seker & Eryigit, 2017) and (Alecakir et al., 2022), results were reported on the re-annotated version.

	Method	TW-2013	IWT	TW-SUNLP
Çelikkaya et al. (2013)	CRF	19.28	-	-
Küçük & Steinberger (2014)	Rule-based	38.01	-	-
Eken & Tantuğ (2015)	CRF	28.53	-	-
Okur et al. (2016)	Reg. Avg. Perc.	48.96	-	-
Seker & Eryigit (2017)*	CRF	67.96	64.96	-
Alecakir et al. (2022)*	BiLSTM-CRF	67.39	-	-
Çarık & Yeniterzi (2022)	BERT	-	-	82.18

Although results in well-written texts are significantly high, when the same methods were applied to noisy text, scores dropped drastically. The CRF system, which was proposed in Şeker & Eryiğit (2012) and achieved over 90% on news articles, obtained only 19.28% on the Twitter dataset with text normalization.

In Seker & Eryigit (2017), the Twitter dataset introduced in (Çelikkaya et al., 2013) was re-annotated to improve the quality of the annotations. With adding new features to CRF to capture the numerical and temporal expressions such as date or money, the score was considerably improved in the re-annotated dataset.

4. NEURAL APPROACHES

In this chapter, we introduce several neural models experimented on all six datasets to address the NER task. We also present our word embeddings and compare them with popular ones. Moreover, we explore several transformer-based models that differ in pre-training corpus and architectural design. Finally, we experimented with CRF and BiLSTM layers on top of transformer-based models to boost their performance.

Neural models, especially transformer-based models, achieved state-of-the-art scores for the NER task on various Turkish datasets (Ozcelik & Toraman, 2022; Safaya et al., 2022). Although transformer architectures outperformed the BiLSTM-CRF model in every dataset (Güneş & Tantuğ, 2018; Güngör et al., 2018), it is still a strong baseline to observe the effectiveness of both transformers-based models and our new approaches. In addition, TW-2013 and IWT datasets have never been evaluated with transformer-based models. Moreover, to compare the effect of including a knowledge base in the same experimental setting, we devoted a separate section to these models.

The architectures used in this part can be divided into three groups; BiLSTM network, transformers-based models, and combining these architectures with a CRF structure. These architectures address this task as a sequence-to-sequence (seq2seq) problem, which generates an output for every token in a given input sentence. The CRF layer was implemented on top of both models to boost their performances. Furthermore, LSTM layers were included in the transformer-based models to explore whether the models can better capture the relations in a sequence.

4.1 BiLSTM Model

Recurrent Neural Networks (RNN) are a special kind of neural network that creates a memory by taking the next token and outputs from the previous timestamp as input to address sequential problems. Long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997) is a variation of RNN introduced to address the vanishing and exploding gradient problem of RNN, as well as capturing relations between words with a longer range. In order to have a wider range of dependencies, different gates determine which information is stored and which is forgotten. Implementation of an LSTM cell is as follows:

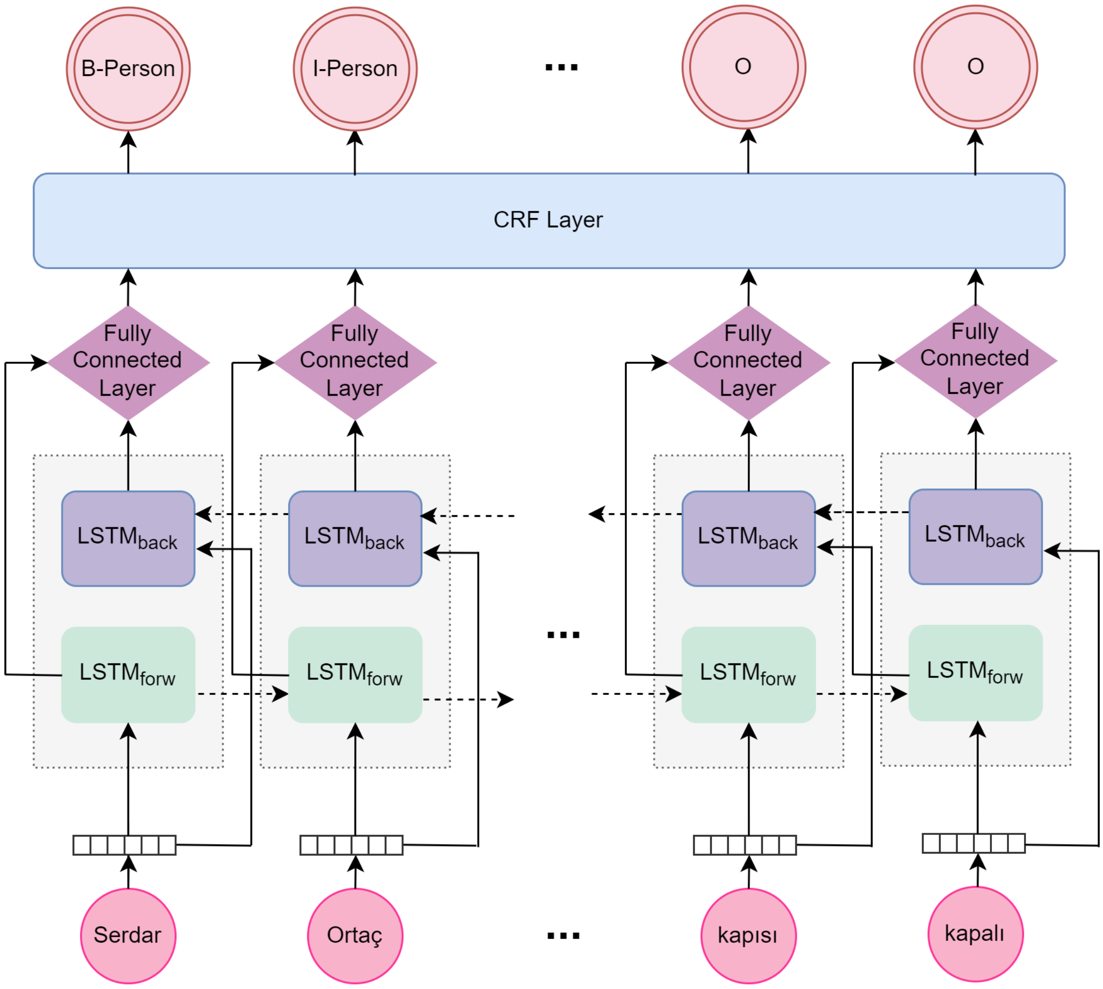
$$\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
g_t &= tanh(W_c \cdot [h_{t-1}, x_t + b_c) \\
c_t &= f_t * c_{t-1} + i_t * g_t \\
h_t &= o_t * tanh(c_t)
\end{aligned} \tag{4.1}$$

where σ is the sigmoid function, x_t is the input at time t, h_t is the hidden, and c_t is the cell state. The input gate is i_t, the forget gate is f_t, the cell is g_t, and the output gate is o_t. W and b's are the weights and biases of the respective gates.

Bidirectional LSTM (BiLSTM) consists of two LSTM layers that process the input sequence in different directions. In the first layer, the flow occurs from the start of a sequence to the end, while in the second layer it is in the opposite direction. The BiLSTM architecture and processing of input with this model are illustrated in Figure 4.1. Each token is initially encoded using a pretrained word embedding. Then, the embedding vectors are fed into the LSTM layers. The output of the layers is concatenated and passed into a fully connected layer. To predict the most probable named entity tags, the output of the fully connected passed into a softmax or CRF layer.

4.1.1 Word Embeddings

Figure 4.1 An illustration of BiLSTM-CRF architecture.



In order to represent the tokens, we experimented with several Turkish word embeddings. The available embedding vectors are trained with mostly Common Crawl[1] data collected from web pages, books, or Wikipedia pages. The BiLSTM-CRF architecture utilized these embedding vectors can compete with transformer-based models in news articles and Wikipedia datasets. However, the performance of transformer-based models on social media datasets is unknown. In order to compare transformer-based models and BiLSTM-CRF architecture in the same experimental setting for social media datasets, we trained Word2Vec and fastText embedding vectors with a large tweet dump.

In this study, we used three different available pretrained word embedding models, namely fastText (Bojanowski et al., 2017), GloVe (Pennington et al., 2014), and Word2Vec (Mikolov et al., 2013), and trained our fastText and Word2Vec embedding models. Details on the word embeddings are reported in Table 4.1.

[1] https://commoncrawl.org/

Table 4.1 Details of word embeddings.

	Training Data	Vocab Size	Dimension	Window Size	Negative Sampling
fastText(Grave et al., 2018)	Formal	2M	300	10	5
Wod2Vec(Güngör & Yıldız, 2017)	Formal	2M	300	5	10
GloVe[2]	Formal	570K	300	-	-
Our fastText	Informal	1.6M	100	10	5
Our Word2Vec	Informal	1.6M	300	10	5

Available Word Embeddings:

- fastText (Grave et al., 2018): It was trained with Common Crawl data and Wikipedia pages. fastText is an effective embedding model since it performs better in morphologically rich languages due to the use of n-gram characters.
- Word2Vec (Güngör & Yıldız, 2017): It was trained with news articles, web pages, and books using the skip-gram model.
- GloVe[2]: It was also trained with Common Crawl data. However, the vocab size of its corpus is relatively small compared to other pretrained word embeddings.

We trained new Word2Vec and fastText models with a large tweet dump collected between June 2020 to June 2021. More than 65 million tweets were selected based on the top 10 trend topics in Turkey. Tweets were fed into the models without any preprocessing step. The skip-gram algorithm was used to train the models with the gensim[3] library. For both models, the window size was chosen as 10, and words encountered less than five times were eliminated. The vector size for the Word2Vec and fastText models was set as 300 and 100, respectively.

4.2 Transformer-based Models

Transformer-based models are deep neural architectures that replaced RNN or CNN layers with feed-forward layers and self-attention. With the attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser & Polosukhin, 2017), the models can capture the contextual information of each token. Also, these large

[2] https://github.com/inzva/Turkish-GloVe

[3] https://radimrehurek.com/gensim/

language models learn the contextual representation of words on an immense corpus with unsupervised tasks such as next-sentence prediction and masked language modeling. These language models can be utilized in different tasks, such as NER, by continuing the training with smaller task-specific datasets on the weights learned on large corpora.

In this study, we fine-tuned the BERT (Devlin, Chang, Lee & Toutanova, 2018) model and its variations, which only contain a stack of encoders that learn text's semantic and syntactic features. Models pre-trained on both multilingual and Turkish corpus from different domains were experimented with our datasets. The details of the BERT models and the variations that were used are listed below.

Turkish Models:

- BERTurk (Stefan, 2020): It was pre-trained on formal texts, i.e., Turkish Wikipedia articles, OSCAR, and OPUS corpus. It is a base model with 12 encoder layers and 768 hidden units on each feed-forward layer.

- BERT_{loodos} (Oluk & Özgur, 2020): Its pre-training data includes informal texts like online blogs and tweets. It is the base model with the same dimensions as BERTurk.

- ConvBERTurk (Stefan, 2020): It is a variation of the BERT model based on ConvBERT (Jiang, Yu, Zhou, Chen, Feng & Yan, 2020) with a different training method. It introduced a span-based dynamic convolution layer into the self-attention of BERT to capture local dependencies. The pre-training corpus is the same as BERTurk.

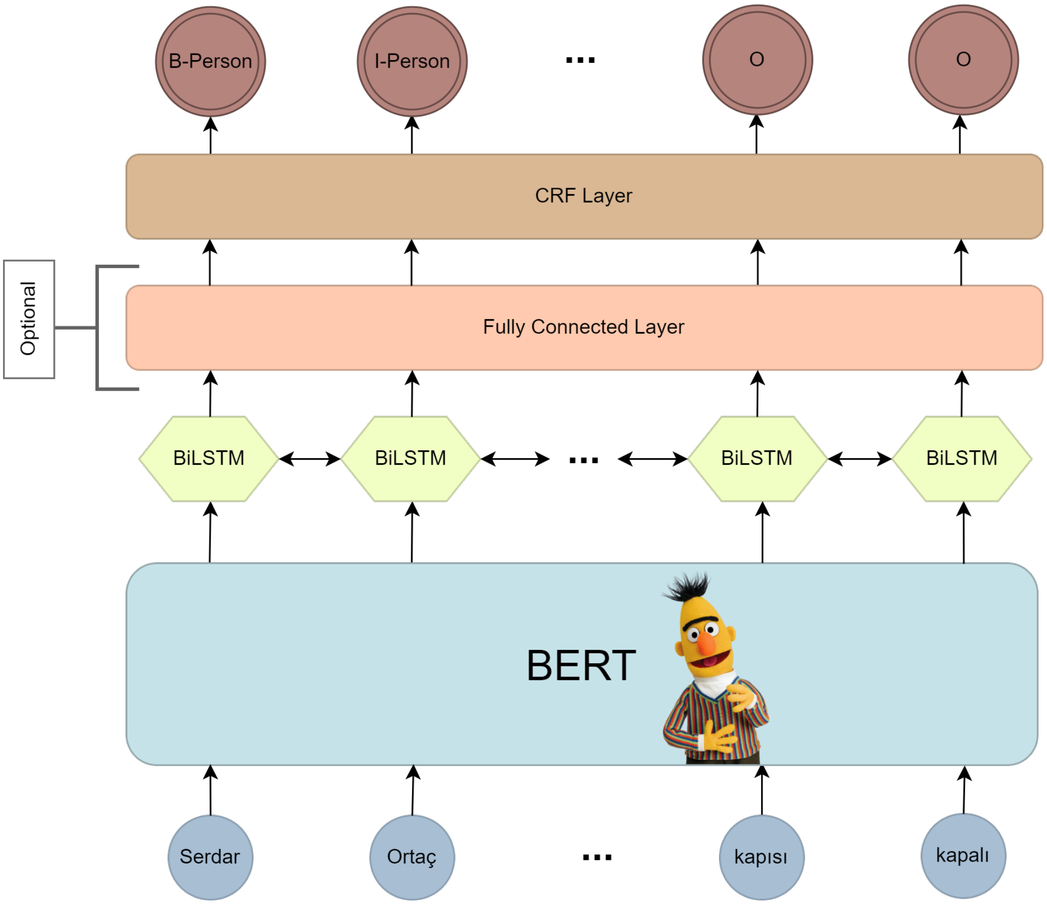
Multilingual Models:

- mBERT (Devlin et al., 2018): The model was pre-trained on Wikipedia pages for 104 languages, including Turkish. The base model was used in this study with 12 encoder layers and 768 hidden units on each feed-forward layer.

- XLM-RoBERTa (XLM-R) (Alexis et al., 2019): The pre-training data of this model collected for 100 languages, including Turkish from the CommonCrawl data. Its difference from the BERT model is excluding the Next Sentence Prediction task in the training process and trained on only masked language modeling objectives.

4.3 CRF Layer

Figure 4.2 An illustration of BERT-BiLSTM-CRF architecture.



The CRF classifier computes the joint probability of all the labels in a sequence to identify the relations at the label level (Lafferty et al., 2001). It also showed competitive performance with neural approaches in Turkish (Şeker & Eryiğit, 2012; Seker & Eryigit, 2017). Hence, instead of using softmax to predict label sequence, we implemented a CRF layer on top of both architectures (i.e., BiLSTM and BERT). The CRF layer calculates the conditional probability $P(y|S)$ as follows:

$$P(y|S) = \frac{\prod_{i=1}^{n} exp(W_{y_i}^T x_i + b_{y_{i-1},y_i})}{\sum_{y' \in \Omega(x)} \prod_{i=1}^{n} exp(W_{y'_i}^T x_i + b_{y'_{i-1},y'_i})} \tag{4.2}$$

where $W_{y_i}^T x_i$ represents the emission and b_{y_{i-1},y_i} represents the transmission scores. $\Omega(x)$ stands for all possible label sequences. Negative log-likelihood loss $_{NLL}(\theta) = -\log p(y|S)$ was used as the loss function in training. To detect the most likely named entity tag sequence, we used the first-order Viterbi algorithm.

The output of the final hidden states of both architectures is passed to the CRF layer after being concatenated. The difference between the BERT model from the

BiLSTM is that BERT splits words that are not in its vocabulary into subwords with its tokenizer. Hence, the model generates output embeddings for each subtoken rather than each token. To implement the CRF layer, we considered only the first sub-token as an input to the next layer and ignored the rest.

Furthermore, to benefit from both transformer-based models and the BiLSTM network, we implemented a BiLSTM layer above the BERT models. As illustrated in Figure 4.2, the output of the BERT model is fed into the BiLSTM layer. As BERT produces an output embedding for each subtoken, again we only fed the first subtoken to the BiLSTM layer. Finally, a CRF layer is applied on top of the BiLSTM layer as in the previous architectures.

4.3.1 Experimental Setup

All neural models were trained with a fixed learning rate of 3x105 using the AdamW (Loshchilov & Hutter, 2018) optimizer. In all experiments, for the BiLSTM-CRF model, the number of epochs was set to 20, and the batch size was 16. For all transformer-based models, the epoch and batch size was set as 10 and 8, respectively. The maximum sequence length size was determined based on the length of input samples in the datasets. Because of the randomization factor in the models' initialization, we trained each model with five different seed values and reported our averaged results.

4.4 Results

In this section, the experiment results of the neural architectures introduced above are presented in Table 4.2 and 4.3. First, we experimented with five different word embeddings on BiLSTM-CRF. Next, the variations of transformer-based models that are trained with multilingual and Turkish corpora were tested on our datasets. Afterward, we evaluated Turkish BERT models by replacing their classification layers with CRF and BiLSTM-CRF layers.

The results of all these architectures are reported in two tables separated as formal and informal. While the average F1 scores in the test sets are provided in the following tables, the precision, recall, and F1 scores in both the test and validation sets are listed in Appendix 7.1.

Table 4.2 The performance of BiLSTM-CRF and transformer-based models on formal datasets (i.e., Milliyet and WikiANN). While the BiLSTM-CRF architecture is separated based on the five different word embeddings, BERT models are divided based on the final layers on top. The average of weighted F1 Scores from 5 runs with different seeds is reported.

Model		Milliyet	WikiANN
BiLSTM-CRF	fastText-FB	90.89±.01	86.57±.00
	GloVe-Inzva	89.89±.00	82.89±.01
	W2V-Gungor	93.63±.00	88.83±.00
	Our fastText	84.69±.01	80.13±.00
	Our W2V	82.10±.01	71.34±.00
BERT	mBERT	87.93±.01	91.52±.00
	XLM-R	89.96±.01	90.12±.00
	BERTurk	95.35±.01	91.34±.01
	BERT$_{loodos}$	94.17±.01	91.45±.00
	ConvBERTurk	95.49±.00	92.06±.00
BERT-CRF	BERTurk	95.72±.00	**93.70**±.00
	BERT$_{loodos}$	95.82±.00	92.77±.01
	ConvBERTurk	**95.94**±.00	93.40±.00
BERT-BiLSTM-CRF	BERTurk	95.87±.00	93.54±.00
	BERT$_{loodos}$	95.46±.00	92.90±.00
	ConvBERTurk	95.76±.00	93.16±.00

In Table 4.2, the evaluation results of BiLSTM-CRF and different transformer-based models on the Milliyet news dataset and WikiANN are reported. The performance of BiLSTM-CRF using various word embeddings is shown in the first five rows. As expected, the W2V-Gungor word embedding model (Güngör & Yıldız, 2017) performed by far the best in Milliyet since its training data includes news articles as an advantage to the fastText-FB (Grave et al., 2018). fastText-FB and GloVe-Inzva achieved significantly higher scores compared to our embedding vectors as the domain of their training data matched with evaluation datasets. The gap between the W2V-Gungor and fastText-FB is decreased in the WikiANN dataset since there are Wikipedia pages in the corpus of fastText-FB. On the other hand, all transformer-based models outperformed BiLSTM-CRF in both datasets, as previously demonstrated in the literature on Turkish NER. Noticeably, the BiLSTM-CRF model with W2V-Gungor embedding vectors outperformed multilingual transformer models and demonstrated comparable performance to the BERT models in Turkish.

The transformer-based models are subdivided according to their variants and pre-training corpora in the table. Although multilingual models obtained poor outcomes in the Milliyet dataset, mBERT performed similarly or even better than some of the Turkish models in the WikiANN. The presence of foreign words in this dataset might increase the effectiveness of multilingual models. Among the three BERT variants, ConvBERTurk achieved the best results in the NER task, demonstrating its superiority over the BERT model as shown in Ozcelik & Toraman (2022).

The performances of the Turkish BERT models were improved by adding a CRF layer, indicating that considering adjacent tokens is effective in deciding the class of the current token. However, an extra layer of BiLSTM did not improve the BERT-CRF model for both datasets.

Table 4.3 The performance of BiLSTM-CRF and transformer-based models on informal datasets (i.e., TW-2013, TW-SUNLP, IWT, and MultiCoNER). While the BiLSTM-CRF architecture is separated based on the five different word embeddings, the BERT model is divided based on the different layers on top. The average of weighted F1 Scores from 5 runs with different seeds is reported.

Model		TW-2013	TW-SUNLP	IWT	MultiCoNER
BiLSTM-CRF	fastText-FB	62.22±.02	68.60±.01	74.47±.03	49.84±.01
	GloVe-Inzva	59.99±.03	69.51±.01	74.17±.02	45.97±.01
	W2V-Gungor	60.83±.01	69.27±.00	74.04±.02	39.33±.01
	Our fastText	48.85±.02	57.19±.01	51.02±.04	43.19±.01
	Our W2V	61.70±.02	65.72±.01	69.47±.01	34.11±.01
BERT	mBERT	59.89±.04	75.40±.00	72.00±.04	44.87±.01
	XLM-R	64.58±.03	77.81±.00	69.65±.07	48.85±.01
	BERTurk	68.38±.04	84.01±.00	85.58±.02	49.95±.01
	BERT_{loodos}	66.75±.01	84.59±.01	83.20±.03	48.83±.01
	ConvBERTurk	69.41±.03	85.04±.01	83.26±.04	54.21±.01
BERT-CRF	BERTurk	**73.52**±.02	85.68±.00	**87.49**±.02	52.31±.01
	BERT_{loodos}	69.38±.02	85.24±.01	85.27±.03	51.98±.00
	ConvBERTurk	70.96±.01	**86.36**±.00	82.86±.01	56.64±.01
BERT-BiLSTM-CRF	BERTurk	63.95±.06	84.29±.01	77.63±.08	51.33±.01
	BERT_{loodos}	63.67±.03	83.99±.01	77.26±.05	50.55±.00
	ConvBERTurk	63.19±.03	85.74±.00	76.94±.03	**57.03**±.00

The F1 scores on informal datasets are presented in Table 4.3. We expected a better performance of the BiLSTM-CRF architecture trained by our word embeddings, as our embedding vectors were trained with tweets. However, both embedding vectors, especially our fastText model, performed poorly compared to other embedding vectors. The reason for the poor results with fastText models may be due to the inadequacy of the smaller size to represent tokens. Also, the smaller size of the training data in our embedding vectors than existing fastText and Word2Vec models may have weakened their performance.

The transformer-based models outperformed BiLSTM-CRF architecture on the informal datasets as well, especially in the Twitter domain. Surprisingly, ConvBERTurk outperformed BERT$_{loodos}$, scoring the highest among other BERT base models. Our expectation was that BERT$_{loodos}$ would perform better on datasets in this domain as its pre-training corpora include informal data. However, the larger vocabulary size of the training corpus of the BERTurk and ConvBERTurk played an important role in the performance of these models. Implementing the CRF layer also improves the transformer-based models on informal datasets. However, the BiLSTM layer could not enhance the BERT-CRF model, or even worsen it, especially for TW-2013 and IWT datasets.

One noticeable point is that the standard deviation between the models with different seeds is remarkably high for the IWT and TW-2013 datasets. The Percent and Time classes in their test sets contain one and two examples, respectively. When different seeds of models miss these samples, their performances are harmed severely, and their standard deviations also increase. These high deviations are observed in the results of the multilingual models and BERTs with the BiLSTM-CRF layer, which may cause a decrease in their performances.

Compared to other informal datasets, results in MultiCoNER are significantly worse. However, as shown in Table A.11, the validation set's results are 30% higher than the test set's scores. The reason for the difference between the results of these two sets is that the samples in the test set are extremely short, with an average of 5 words. Such short sentences hinder the performance of transformer-based models, which are hungry for context.

5. KNOWLEDGE-BASED APPROACHES

In this chapter, we present our methodologies for the NER task, which enriched existing models with external knowledge. We explain our methods to construct external knowledge by utilizing both entity linking approach and contextual information.

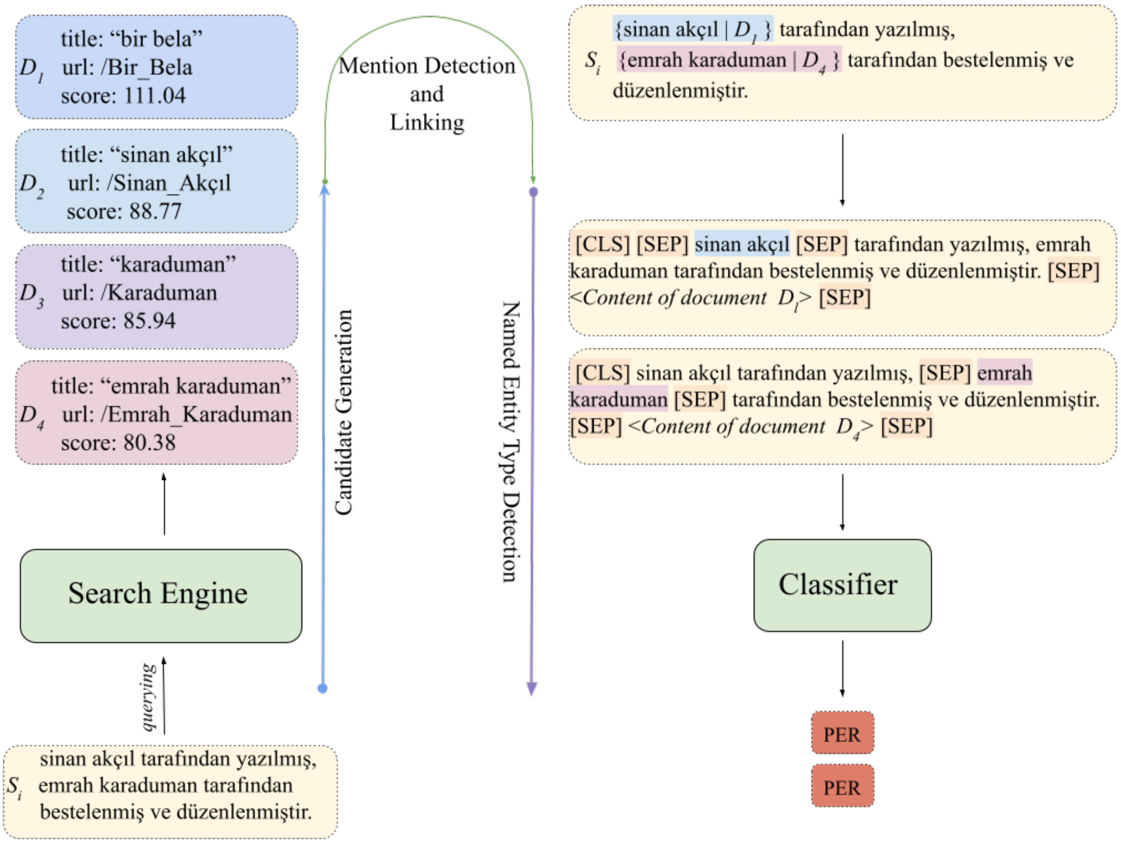
The neural models described in the previous section achieve remarkable results. However, in instances where context is absent, such as in tweets and search queries, the performance of these architectures degrades as they are heavily dependent on context. Furthermore, the number of unseen entities encountered in the informal dataset is significantly higher compared to formal ones. Besides many unseen entities, the named entity classes in the informal datasets enlarged to include complex and emerging entities like book or movie titles.

In order to address these challenges, integrating information from gazetteers and knowledge bases has shown improvement in the NER task over the years (Fetahu et al., 2021; Lin et al., 2019). Besides, models might leverage external information to learn about rare and unknown entities (Wang, Qu, Chen, Shen, Zhang, Zhang, Gao, Gu, Chen & Yu, 2018). Hence, we proposed a new pipeline that extracts additional knowledge from a knowledge base to integrate transformer-based models in an unsupervised way. We also adopted a recent approach (Wang et al., 2021) that utilized semantically similar contents as external knowledge in our pipeline.

5.1 $\mathrm{EL}_{\mathrm{BERT}}$ & $\mathrm{EL}_{\mathrm{MultiBERT}}$

In the first approach, we aimed to provide additional information to the BERT model from Wikipedia pages by detecting the pages of possible named entities. The proposed pipeline is illustrated in Figure 5.1. The overall system can be examined

Figure 5.1 An illustration of both EL$_{\text{BERT}}$ and EL$_{\text{MultiBERT}}$ pipelines.


title: “bir bela”
D_1 url: /Bir_Bela
score: 111.04
title: “sinan akçıl”
D_2 url: /Sinan_Akçıl
score: 88.77
title: “karaduman”
D_3 url: /Karaduman
score: 85.94
title: “emrah karaduman”
D_4 url: /Emrah_Karaduman
score: 80.38
Search Engine
querying
S_i sinan akçıl tarafından yazılmış, emrah karaduman tarafından bestelenmiş ve düzenlenmiştir.
Mention Detection and Linking
Candidate Generation
Named Entity Type Detection
S_i {sinan akçıl | D_1 } tarafından yazılmış, {emrah karaduman | D_4 } tarafından bestelenmiş ve düzenlenmiştir.
[CLS] [SEP] sinan akçıl [SEP] tarafından yazılmış, emrah karaduman tarafından bestelenmiş ve düzenlenmiştir. [SEP] <Content of document D_1> [SEP]
[CLS] sinan akçıl tarafından yazılmış, [SEP] emrah karaduman [SEP] tarafından bestelenmiş ve düzenlenmiştir. [SEP] <Content of document D_4> [SEP]
Classifier
PER
PER

in three components. The first section is Candidate Generation, where an input example x with n tokens, $x = \{x_1, ..., x_n\}$, is searched in Wikipedia articles to obtain a list of relevant documents. In the Mention Detection part, possible named entities are identified by searching for the input sample in the retrieved documents and detected possible entities are matched with the corresponding Wikipedia page. In the final step, the input sample and relevant Wikipedia page are fed into the BERTurk model to generate named entity predictions for each possible entity detected in the second section.

5.1.1 Candidate Generation

Wikipedia is an open-source and constantly evolving knowledge base. It also contains mention hyperlinks, namely wiki anchors, in the page contents that linked to the entities' own Wikipedia pages. To efficiently search for inputs in large volumes of documents, we indexed the Wikipedia pages using the ElasticSearch (ES)[1] search engine with default settings. ES is a fast and open-source search engine that performs searching at a document level. We represent each document D_i with the following fields:

- title: Title of a Wikipedia page.
- content: Page content of a Wikipedia page.
- referred_by: List of text spans, including title, showing how other pages refer to this document. They are collected from other pages using wiki anchors. For example, the United States is referred to as U.S., USA, or The United States of America on other Wikipedia pages.
- interwikies: List of wiki anchors that are mentioned on the Wikipedia page. For instance, there are hyperlinks on the United States' page to the New York City and Washington D.C.'s Wikipedia pages.
- all_content: Concatenated version of all fields shown above.

Input samples were used as queries to obtain relevant Wikipedia pages. We retrieved the most relevant 200 pages for title, referred_by, interwikies, and all_content fields by calculating the similarity score for each document using the BM25 algorithm. The reason for retrieving a large number of documents is to not miss a relevant Wikipedia page. The collected search results were pooled to find possible entities mentioned differently from their Wikipedia title.

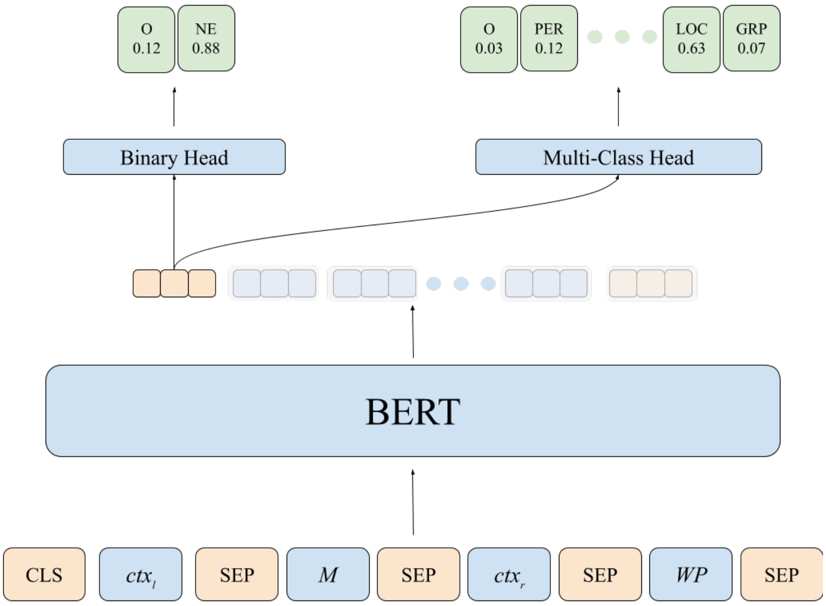
5.1.2 Mention Detection

In this section, the pooled documents for each input sample were searched over the text spans of the input samples to find possible entity mentions. Identifying the boundaries of a complex entity can be difficult by considering only its contextual representation. For example, *Star Wars: Episode II - Attack of the Clones*, a sequel to the Star Wars series, is a named entity found in the WikiANN dataset. Although there are numerous documents related to Star Wars, it might be hard to detect all the tokens of this movie's title by only giving Star Wars-related content. On the other hand, this sequel has its own Wikipedia page[2], therefore, it might be

[1] https://www.elastic.co/

[2] `https://en.wikipedia.org/wiki/Star_Wars:_Episode_II_%E2%80%93_Attack_of_the_Clones`

Figure 5.2 An illustration of the two-headed classifier of the $\text{EL}_{\text{MultiBERT}}$ pipeline.



helpful to first identify possible entities by searching Wikipedia for phrases in inputs. Moreover, using both title and referred_by fields when searching documents allows us to detect synonyms or abbreviations for entities. For instance, since the United States of America is referred to as USA or U.S. on other pages, by collecting this information, we can also detect this entity under different names. Thus, we provide external information to the transformer-based models by mapping possible mentions to relevant Wikipedia content. In order to match the candidate documents to the possible named entities, we applied a straightforward algorithm.

The algorithm looks for an exact match between input sentences and pooled documents by comparing each document's referred_by field with the input. The referred_by and title fields of each document are compared with the input to determine whether there is an exact match. By iterating the pooled documents for each input sample, we generated a list of possible named entities that mapped to specific Wikipedia pages. If there was an overlap between the matched entities, we selected the longer one. For matches of the same length, the document with the highest relevance score was taken into account. The relevance score from different field retrievals was added together after pooling to determine the final relevance score.

5.1.3 Named Entity Type Detection

To detect the type of the named entity with external context, we implemented two different methods separated based on the classifier layer. The first approach, called EL_{BERT}, is a vanilla BERTurk model with a single classification layer that was trained with the input and corresponding Wikipedia page to predict the named entity class of a mention. The other method, called $EL_{MultiBERT}$, has two classifier heads. The two-headed architecture has shown in Figure 5.2. While one head is responsible for detecting the named entity type of the mention, the other head resolves an auxiliary task that determines if the mention is an entity. The prediction of the head that detects the classes was ignored if the binary head was predicted as non-entity. The final loss function is calculated by adding the losses of the two heads. Instead of just giving the input sentence and generating the named entity classes for each token, each possible entity that matched with a Wikipedia entry in each input sentence was fed into both architectures to predict the class of the possible entity. The input sentence and mapped article content were represented as follows:

$$\texttt{[CLS]}\, ctx_l\, \texttt{[SEP]}\, M\, \texttt{[SEP]}\, ctx_r\, \texttt{[SEP]}\, WP\, \texttt{[SEP]}$$

where the input sentence and Wikipedia content were separated with a special `[SEP]` token. In order to emphasize the possible named entity, we add `[SEP]` tokens before and after the mention. Mention is represented with M, mapped Wikipedia page is represented with WP, and the tokens before and after the mention are represented with ctx_l and ctx_r, respectively.

5.1.4 Experimental Setup

We used the latest Turkish Wikipedia dump, which was released on October 1, 2022. The dump contains 1,322,956 entities in total, but after eliminating the redirect pages, we have 818,115 pages left. Of the remaining pages, there are 3311 pages with no content.

Since BERT has a 512 token limit, long input samples, especially from the news domain, have no room for external context. Hence, we divided the long samples, specifically those longer than 60 tokens, into sub-samples. If they are still not short enough, commas are used to separate them into smaller units. Then, these sub-samples were fed into the BERT model after concatenating with related content.

As in the neural models, we trained the BERTurk models with a fixed learning rate of $3x10^5$ using the AdamW (Loshchilov & Hutter, 2018) optimizer. In all experiments, the number of epochs was set to 8, and the batch size was 4. The maximum sequence length was expanded to 512, which is the maximum value for BERT and its variations since we added external information. Again, we trained each model with five different seed values and reported our averaged results because of the randomization.

5.1.5 Results

In this section, we present the search results of ES in the Candidate Generation section. We also evaluate how many named entities we were able to detect in the Mention Detection section. Furthermore, we report the results of the EL_{BERT} and $EL_{MultiBERT}$ pipelines. Moreover, we compare the performance of these pipelines with the BERTurk and BERTurk-CRF models.

5.1.5.1 Search Results in ElasticSearch

In Table 5.1, we demonstrate the search results in ES for each dataset by averaging the title, referred_by, interwikies, and all_content fields. The first column shows, on average, how many of the queried input samples returned empty. The average number of documents retrieved for a sample from each field is displayed in the second column. The third column gives the average number of documents for each sample after combining all documents uniquely from each field.

A substantial amount of queries in Milliyet, TW-2013, and MultiCoNER datasets returned empty. Although the average sample length in the Milliyet dataset is high, there are extremely short examples, such as part of dialogues, especially in the training set. Tweets in the TW-2013 and samples in the test set of MultiCoNER are also significantly short. Furthermore, the noisy nature of the tweets and a high number of phrases from other languages in the MultiCoNER dataset increase the amount of empty returning queries. Since the number of named entities and average sample length in the TW-SUNLP is high for a social media dataset, its search results were not affected as others. Moreover, the number of named entities in the IWT dataset is relatively low compared to others, resulting in fewer documents being retrieved for each field.

Table 5.1 Statistics of search results. The first column displays the average of the query results returning empty for each field. The average number of documents received in each field, namely title, referred_by, interwikies, and all_content, is shown in the second column. The last column shows the average number of documents after pooling for each dataset.

Dataset	Empty Queries	Retrieved Docs	Pool Size
Milliyet	46.17	190.42	564.68
WikiANN	9.5	185.88	492.86
TW-2013	57.75	179.86	511.88
TW-SUNLP	1.67	198.27	563.65
IWT	1.92	97.08	284.14
MultiCoNER	113.67	192.66	541.20

5.1.5.2 Effectiveness of Mention Detection

To evaluate the named entity detection performance of searching over Wikipedia pages, we calculated recall scores by checking how many of the possible entities we matched in this part are actually named entities. The scores are shown in Table 5.2 for each dataset.

Table 5.2 Recall scores that are calculated according to detected entity mentions by the proposed algorithm for each dataset.

Dataset	Train	Validation	Test
Milliyet	66.15	67.24	65.86
WikiANN	80.22	80.02	80.07
TW-2013	57.53	63.33	53.01
TW-SUNLP	62.64	62.67	62.19
IWT	76.71	78.90	80.34
MultiCoNER	98.89	98.85	89.84

The system can detect the majority of named entities in the MultiCoNER and WikiANN datasets since they were mostly derived from Wikipedia articles. Finding named entities by searching only exact matches in Twitter datasets is limited due to the irregularities in tweets. Although texts in the news dataset are well-written, most of the named entities belong to the Person class. Since not all people in the news are public figures, not every example in this class has a Wikipedia article. Therefore, the number of named entities that were captured remained modest for the Milliyet dataset. Recall scores for the IWT dataset are surprisingly high. This may be due to the fact that fewer spelling errors and shorter queries help ES performs better.

5.1.5.3 Results of $\mathbf{EL_{BERT}}$ and $\mathbf{EL_{MultiBERT}}$

The F1 Scores of EL_{BERT} and $EL_{MultiBERT}$ architectures are reported in Tables 5.3 and 5.4, separated as formal and informal datasets. $\mathrm{EL_{BERT}}$ and $\mathrm{EL_{MultiBERT}}$ performed poorly in all datasets except MultiCoNER. Since these approaches were trained to detect the type of possible entities that matched in the Mention Detection part, the maximum detection possibility of these models is bounded by the number of matched entities demonstrated in Table 5.2. The unknown named entities, unfortunately, were ignored.

Table 5.3 The performance of EL_{BERT} and $EL_{MultiBERT}$ on formal datasets (i.e., Milliyet and WikiANN). For comparison, the BERTurk and BERTurk-CRF models are also included. The average of weighted F1 Scores from 5 runs with different seeds is reported.

	Milliyet	WikiANN
BERTurk	95.35±.01	91.34±.01
BERTurk-CRF	**95.72**±.00	**93.70**±.00
$\mathrm{EL_{BERT}}$	54.95±.00	62.33±.00
$\mathrm{EL_{MultiBERT}}$	54.72±.00	62.09±.00

Table 5.4 The performance of EL_{BERT} and $EL_{MultiBERT}$ on all informal datasets. For comparison, the BERTurk and BERTurk-CRF models are also included. The average of weighted F1 Scores from 5 runs with different seeds is reported.

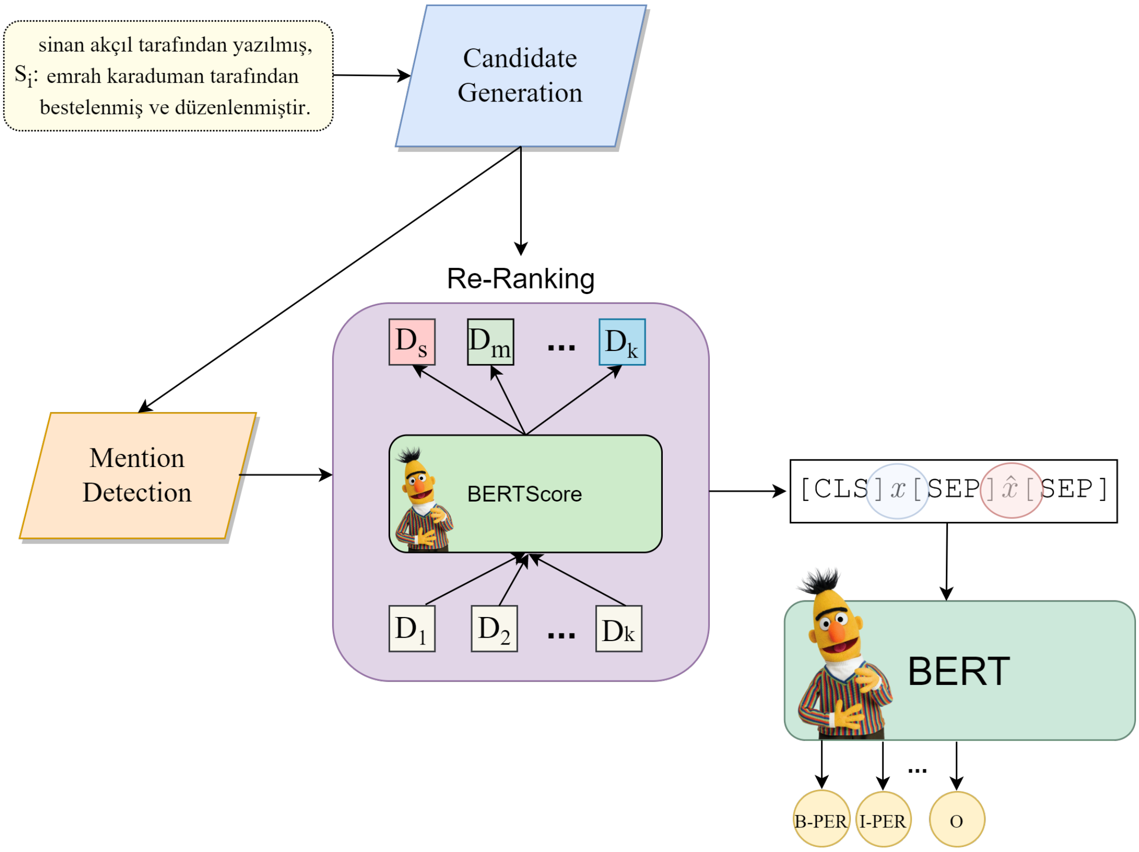
	TW-2013	TW-SUNLP	IWT	MultiCoNER
BERTurk	68.38±.04	84.01±.00	85.58±.02	49.95±.01
BERTurk-CRF	**73.52**±.02	**85.68**±.00	**87.49**±.02	52.31±.01
$\mathrm{EL_{BERT}}$	35.67±.08	53.42±.00	67.86±.02	68.07±.03
$\mathrm{EL_{MultiBERT}}$	35.78±.06	53.32±.00	68.42±.03	**69.32**±.01

Due to the noisy nature of tweets and many unknown person names in the news, the number of matches was moderate in these datasets. Moreover, the Wikipedia content of some of the matched entities might return empty. Table A.13 in the Appendix lists the number of empty pages for each dataset. These factors led to the failure of these models in almost all datasets. On the other hand, $EL_{MultiBERT}$ improved the transformer-based models significantly on the MultiCoNER dataset since its test set has a large number of unique and complex named entities and almost no context.

5.2 $\text{EL}_{\text{Semantic}}$

Since it is not always possible to find the exact matches in Wikipedia, their performances are limited to the number of possible entities that the matching algorithm can capture. Moreover, we cannot leverage the BERT's skills for the remaining input as it is only forced to predict possible entities. Another limitation of EL_{BERT} and $EL_{MultiBERT}$ is that Wikipedia contains articles about common objects like *book* or *house*. The noises added because of these pages might degrade the model's performance (Chiu & Nichols, 2016). Furthermore, the BM25 algorithm is designed to rapidly find structurally close documents fast, ignoring contextual similarity. Hence, using articles returned from ES directly may expose the model to irrelevant pages.

Figure 5.3 An illustration of the $\text{EL}_{\text{Semantic}}$ pipeline.


sinan akçıl tarafından yazılmış,
S_i: emrah karaduman tarafından
bestelenmiş ve düzenlenmiştir.
Candidate Generation
Re-Ranking
D_s
D_m
...
D_k
Mention Detection
BERTScore
D_1
D_2
...
Dk
[CLS] x [SEP] $\hat{x}$ [SEP]
BERT
...
B-PER
I-PER
O

To alleviate the limitations of EL_{BERT} and $EL_{MultiBERT}$, we introduced a new pipeline that utilizes semantically closer content as an external knowledge to provide the model with information about entities that cannot be found in the Mention Detection and eliminate the irrelevant pages. The new approach, called EL_{Semantic}, is illustrated in Figure 5.3. The Candidate Generation and Mention Detection parts remained the same as in the previous methods. In this pipeline, we included the Re-ranking section to emphasize semantically closer pages so that even if the exact matches cannot be found in Wikipedia, we can provide sentences that discuss similar topics as additional information. In the Re-ranking, the search results were re-ordered by comparing the contextual representation of the input and retrieved pages. To measure the semantic similarity of the input sentences and documents, we calculated BERTScore (Zhang, Kishore, Wu, Weinberger & Artzi, 2019) as proposed in (Wang et al., 2021). Furthermore, Wikipedia pages that mapped to a possible named entity in the Mention Detection were given additional weight in the re-ranking process to prevent missing a possible entity. The classification of named entity classes with the BERTurk model was also changed. Now, instead of just predicting possible entities, BERT generates a prediction for each token in the input sample.

5.2.1 Re-ranking

To find contextually similar pages, we followed the method proposed by Wang et al. (2021). They suggested adding semantically relevant context to the transformer-based model by concatenating it with the input sentence. Since the objective of search engines retrieving documents at high speed, some of the search results with high relevance scores might be irrelevant to the query. Hence, in order to find semantically similar contents, they re-ranked the retrieved documents from a search engine based on their contextual closeness. To measure the similarity of the input sentence and document, BERTScore (Zhang et al., 2019) was employed since it is used as an evaluation metric for text generation tasks to evaluate how semantically similar are the two sentences. Hence, a document has a higher BERTScore when it is semantically more similar to the input sentence.

Since not all named entities have an entry on Wikipedia, we might increase the effectiveness of adding external knowledge to transformer-based models by focusing on pages that are semantically similar to the input sample among the retrieved documents. As a contribution to the Wang et al. (2021)'s work, we emphasize the pages that were mapped to potential entities in the Mention Detection part so that the synonyms and abbreviations of the entities were also taken into account. While measuring the semantic similarity, we calculated BERTScore between each input and documents in its pool. First, each token from input sentences and the first paragraph of documents are represented with contextual embedding vectors generated from the BERT model. We preferred to use the BERTurk model to create embeddings as it was pre-trained with Turkish texts. BERTScore first calculates the cosine similarity score between each token of the two sentences to be compared. After each token in the document is matched with the token that has the highest cosine similarity score in the input sentence, the recall score is computed by summing the cosine scores between the matched tokens. Similarly, the precision score is calculated by summing the cosine similarity scores of each token in the input sentence that matches the tokens in the document. For input sentence $x = \{x_1, ..., x_n\}$ and retrieved document $\hat{x} = \{\hat{x}_1, ..., \hat{x}_m\}$, BERTScore's precision, recall and f1 metrics are calculated as follows:

$$\textit{Recall} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^{\mathrm{T}} \hat{x}_j \qquad \textit{Precision} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^{\mathrm{T}} \hat{x}_j \qquad F1 = 2\frac{P \cdot R}{P + R} \tag{5.1}$$

Since pages that matched with phrases in the input sentences are likely to be named entities, more weight has been assigned to the matched pages while re-ranking. For each input sentence, the pooled documents in the Candidate Generation subsection were re-ranked according to calculated f1 scores as in Formula 5.1 and added additional scores based on whether the page matched a part of the input sample. The first 5 or 6 pages in the pool, which were re-ranked according to the semantic relevance with the input samples, were selected as external content.

5.2.2 NER Classifier

In order to take advantage of BERT's capabilities on named entities that cannot be captured in the Mention Detection part, we changed the input representation for the BERT model. First, we concatenated the top six Wikipedia pages that were selected as external knowledge after the re-ranking process. The external contexts were separated from each other by the special `[SEP]` token. Again, the input sentence and the external knowledge were merged with the `[SEP]` token. The final representation of the input before it is fed into the model is given below.

$$\texttt{[CLS]}\, x\, \texttt{[SEP]}\, \mathit{WP_1}\, \texttt{[SEP]} \ldots \texttt{[SEP]}\, \mathit{WP_6}\, \texttt{[SEP]}$$

where x is the input sample, $WP_{\text{1-6}}$ are the first paragraph of the six most relevant Wikipedia pages after re-ranking the search results. The BERTurk model was trained with these new inputs to predict the named entity classes of each token in input sentences.

5.2.3 Results

In this section, we present the results of the EL$_{\text{Semantic}}$ and compared them with our previous pipelines and BERTurk and BERTurk-CRF models.

The F1 Scores of EL$_{\text{Semantic}}$ and previous architectures are reported in Tables 5.5 and 5.6, separated as formal and informal datasets. Our EL$_{\text{Semantic}}$ pipeline outperformed the BERTurk model on almost all datasets. Adding semantically relevant information resulted in further improvement, especially in context-deprived datasets. However, none of the methods exploiting external context contributed to the BERT model as much as the CRF layer, not surprisingly, except for the MultiCoNER dataset.

Table 5.5 The performance of EL$_{\text{Semantic}}$ on formal datasets (i.e., Milliyet and WikiANN). For comparison, the EL$_{\text{BERT}}$, EL$_{\text{MultiBERT}}$, BERTurk, and BERTurk-CRF models are also included. The average of weighted F1 Scores from 5 runs with different seeds is reported.

	Milliyet	WikiANN
BERTurk	95.35±.01	91.34±.01
BERTurk-CRF	**95.72**±.00	**93.70**±.00
EL$_{\text{BERT}}$	54.95±.00	62.33±.00
EL$_{\text{MultiBERT}}$	54.72±.00	62.09±.00
EL$_{\text{Semantic}}$	94.04±.01	92.82±.00

Table 5.6 The performance of EL$_{\text{Semantic}}$ on all informal datasets. For comparison, the EL$_{\text{BERT}}$, EL$_{\text{MultiBERT}}$, BERTurk, and BERTurk-CRF models are also included. The average of weighted F1 Scores from 5 runs with different seeds is reported.

	TW-2013	TW-SUNLP	IWT	MultiCoNER
BERTurk	68.38±.04	84.01±.00	85.58±.02	49.95±.01
BERTurk-CRF	**73.52**±.02	**85.68**±.00	**87.49**±.02	52.31±.01
EL$_{\text{BERT}}$	35.67±.08	53.42±.00	67.86±.02	68.07±.03
EL$_{\text{MultiBERT}}$	35.78±.06	53.32±.00	68.42±.03	69.32±.01
EL$_{\text{Semantic}}$	73.49±.01	84.77±.01	86.83±.01	**69.67**±.01

The $\text{EL}_{\text{Semantic}}$ lagged behind the BERTurk model only in the news articles dataset. Since external context is appended to the end of the input, long samples in the Milliyet dataset had to be split to fit the BERT model. Dividing the sentences hurt the models' performances since it caused a loss in the context. This finding indicates that the model benefits more from its own context than from external knowledge of news articles.

The performance of external context in the TW-SUNLP is different than the other Twitter dataset since the average sample length in the TW-SUNLP dataset is significantly longer compared to other social media datasets. For example, while the average number of words is 9 for TW-2013, this number increased to 25 for TW-SUNLP. The longer samples also increased the amount of context, thus reducing the need for additional information. Hence, improvement by including external information is limited in the TW-SUNLP dataset with external knowledge.

6. DISCUSSION

In this chapter, we compare the performances of the neural and knowledge-based approaches to measure the effectiveness of adding external knowledge to the transfer-based models. We also revealed the strengths and weaknesses of our proposed models by examining them from different aspects. Besides comparing the effectiveness of the models, we also discuss their efficiency in terms of training and testing time.

Adding external knowledge achieved outstanding results over the vanilla BERT model, especially on noisy and short datasets, namely TW-2013 and MultiCoNER. The scarcity of matching entities due to the many irregularities led to poor performance of the $\mathrm{EL}_{\mathrm{BERT}}$ and $\mathrm{EL}_{\mathrm{MultiBERT}}$ architectures on the TW-2013 dataset. In general, the small number of entities that matched a Wikipedia entry in Mention Detection, except for the MultiCoNER dataset, caused these two architectures to fail in all datasets. The improvement with $\mathrm{EL}_{\mathrm{Semantic}}$ on formal datasets was limited including IWT and TW-SUNLP because the structures of these two datasets are similar in terms of writing style and sample length, respectively. Hence, the gain from external knowledge is higher when texts are shorter and noisy.

First, we compared the performance of the $\mathrm{EL}_{\mathrm{Semantic}}$ to BERTurk in each named entity class to analyze whether adding external information to BERTurk improves its effectiveness, especially for complex named entities. In Table 6.1, we reported the results of BERTurk, BERT-CRF, and $\mathrm{EL}_{\mathrm{Semantic}}$ on each entity class. In Milliyet, WikiANN, IWT, and TW-SUNLP, the improvement in the PLO classes is limited, indicating that additional context is not helpful in the examples BERTurk missed. In fact, a CRF layer is more effective for these datasets, especially in PLO classes. On the other hand, including external information improved BERTurk's scores in all classes for noisy and short texts, particularly Creative-Work, Groups, and TV-Show. The improvement in these complex entities is even higher compared to the PLO classes. Considering that the average sentence length in the test set of the MultiCoNER dataset is five, our approach outperformed since the BERTurk and BERTurk-CRF have no information to predict the entities. Therefore, for examples containing complex entities and a lack of context, leveraging external information is significantly effective.

Table 6.1 The performances of BERTurk, BERTurk-CRF, and $\mathrm{EL}_{\mathrm{Semantic}}$'s on each entity class in six Turkish datasets. The average of weighted F1 Scores from 5 runs with different seeds is reported.

Dataset	Entity Class	$\mathrm{EL}_{\mathrm{Semantic}}$	BERTurk	BERTurk-CRF
Milliyet	Person	0.96	0.97	0.97
	Location	0.94	0.95	0.96
	Organization	0.91	0.93	0.93
WikiANN	Person	0.95	0.94	0.96
	Location	0.93	0.92	0.94
	Organization	0.90	0.88	0.91
IWT	Person	0.90	0.91	0.93
	Location	0.88	0.89	0.86
	Organization	0.83	0.82	0.83
	Money	0.68	0.63	0.68
	Date	1.00	0.92	1.00
	Time	0.70	0.41	0.93
	Percent	1.00	1.00	1.00
TW-2013	Person	0.76	0.69	0.74
	Location	0.76	0.70	0.77
	Organization	0.73	0.69	0.72
	Money	0.54	0.56	0.25
	Date	0.61	0.65	0.70
	Time	0.61	0.61	0.45
	Percent	0.60	0.20	0.40
TW-SUNLP	Person	0.91	0.90	0.92
	Location	0.77	0.76	0.76
	Organization	0.83	0.82	0.83
	Money	0.85	0.88	0.87
	Time	0.89	0.89	0.89
	Product	0.50	0.50	0.51
	TV-Show	0.49	0.43	0.53
MultiCoNER	Person	0.75	0.61	0.63
	Location	0.65	0.51	0.55
	Group	0.65	0.40	0.42
	Corporation	0.70	0.50	0.52
	Product	0.77	0.57	0.57
	Creative Work	0.65	0.39	0.42

Furthermore, to investigate whether adding external information can be more effective than the BERTurk model in the scenario where we can find related content about all named entities, we evaluated the $\mathrm{EL}_{\mathrm{BERT}}$ and BERTurk models only on named entities detected in the Mention Detection part. The results are shown in

Table 6.2. The right side of the table demonstrates the performance of the models across all instances of the datasets. On the left, we only considered samples that have a detected named entity. If a selected sample contains an undetected named entity, the undetected entity is considered as an Other class.

Table 6.2 The performances of BERTurk and $\mathrm{EL}_{\mathrm{BERT}}$ models only on named entities detected in the Mention Detection part. The results of these models on entire datasets are also depicted on the left side of the table for comparison.

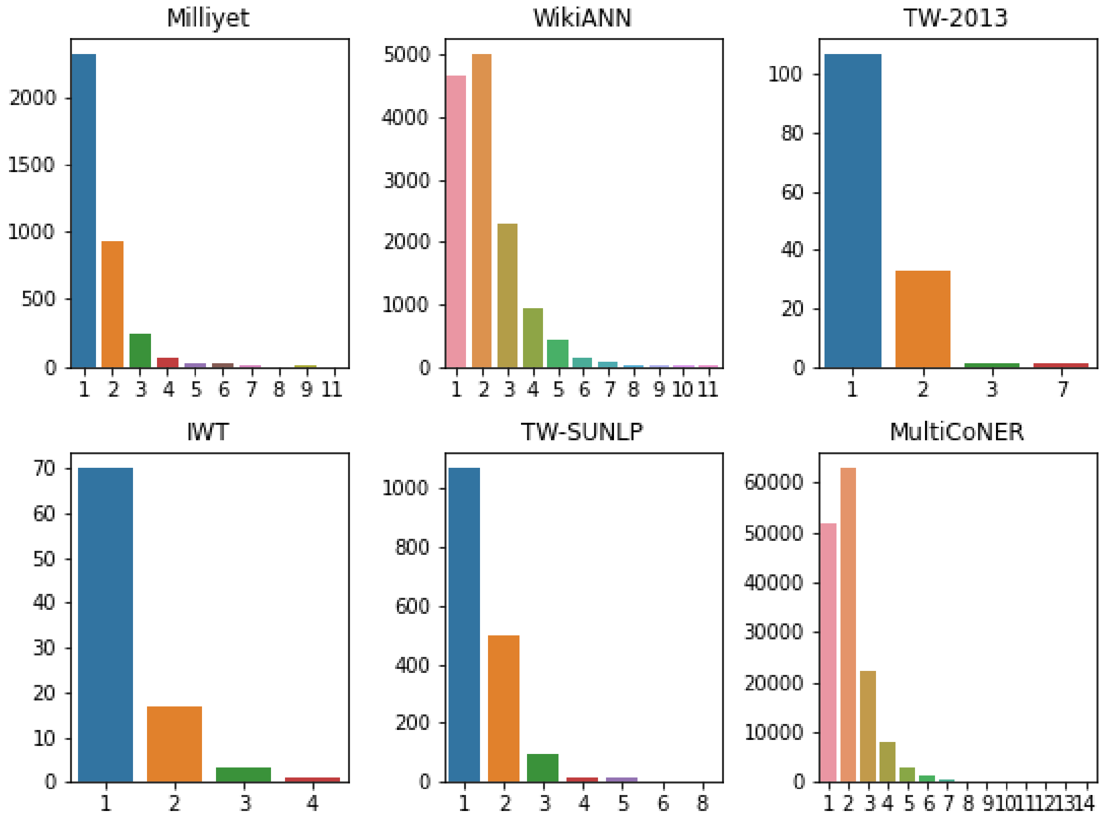
	Full Datasets		Only Detected Entities	
Dataset	BERTurk	$\mathrm{EL}_{\mathrm{BERT}}$	BERTurk	$\mathrm{EL}_{\mathrm{BERT}}$
Milliyet	95.35	54.95	97.48	80.17
WikiANN	91.34	62.33	91.98	93.35
TW-2013	68.38	35.67	90.08	73.87
TW-SUNLP	84.01	53.42	90.03	83.30
IWT	85.58	67.86	91.14	86.67
MultiCoNER	49.95	68.07	59.45	78.10

In MultiCoNER and WikiANN datasets, $\mathrm{EL}_{\mathrm{BERT}}$ outperformed the BERTurk model, however, the score on MultiCoNER for $\mathrm{EL}_{\mathrm{BERT}}$ is lower than the other datasets. The foreign phrases and errors due to the automatic annotation might have limited its performance. Although $\mathrm{EL}_{\mathrm{BERT}}$ achieved significantly higher scores compared to all datasets by focusing on detected entities, it still lagged behind the BERTurk model. For the news dataset, the context of the original sentence is more beneficial than external knowledge. The mapped Wikipedia pages may even add some noise to the input. In the Twitter datasets, BERTurk unexpectedly showed better performance than $\mathrm{EL}_{\mathrm{BERT}}$. However, using a different meaning from the Wikipedia page in the original sentence may have caused some entities to be misclassified. Even though $\mathrm{EL}_{\mathrm{BERT}}$ could not improve the BERTurk on Twitter datasets, including additional context is still more effective in short examples with complex entities and insufficient context.

In addition, we examined the effect of integrating external knowledge into the BERTurk model on long named entities. To measure the effectiveness of the external knowledge in these long entities, we analyzed F1 scores of BERTurk, BERTurk-CRF, and $\mathrm{EL}_{\mathrm{Semantic}}$ on entities of specific lengths. First, we calculated the number of entities of each length for all datasets and depicted them in Figure 6.1. The majority of the named entities are clustered in the 1 to 3 range, as expected. Also, almost all datasets have several long named entities, especially Milliyet, WikiANN, and MultiCoNER.

We present the performance of models at different lengths of entities in Figure 6.2.

Figure 6.1 The number of named entities grouped by their length for each dataset.



In the IWT and formal datasets, our method was able to detect most of the long entities. Compared to BERTurk-CRF, including external knowledge showed a more consistent performance in long entities. While our approach achieved outstanding results in short entities where the majority of the samples were gathered, the medium-length entities are the ones that hurt our pipeline the most in the formal datasets. For the Twitter datasets, external information improved the transformer-based models in longer entities compared to BERTurk and BERTurk-CRF. However, the longest entities of these datasets were not captured by our model. Therefore, we took a closer look at these missed samples, which we have listed below for both datasets.

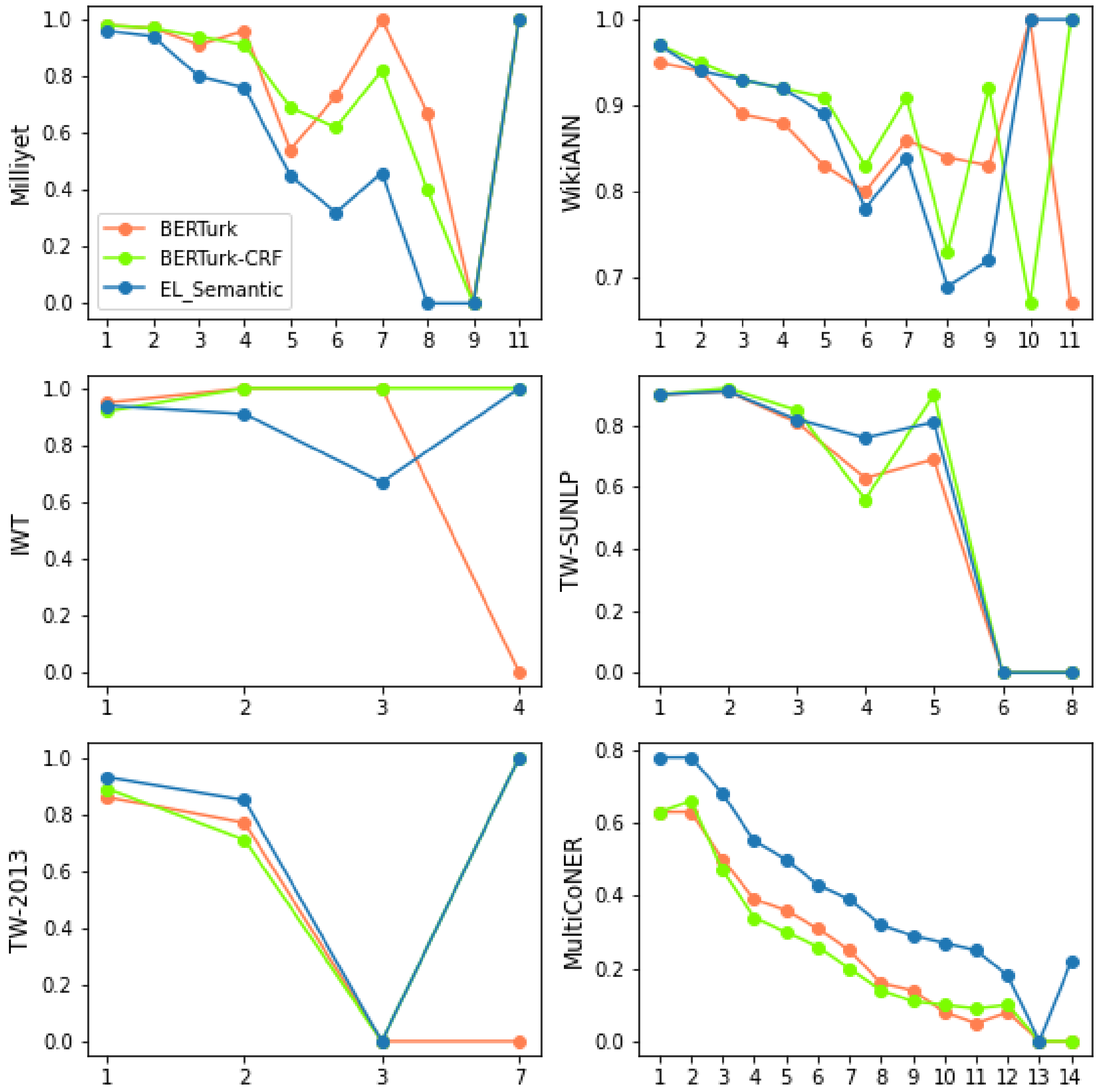
From TW-2013:

- Ali Sami Yen Türk Telekom Arena Stadı / *Ali Sami Yen Turk Telekom Arena Stadium*: Location

From TW-SUNLP:

- Paris Saint-Germain Fan Token: Money

Figure 6.2 Performances of EL$_{\text{Semantic}}$, BERTurk, and BERTurk-CRF in different entity lengths. The y-axis shows the F1 score of the models, while the x-axis shows the various entity lengths.



- Karesi Belediyesi Kültür ve Sanat Evi / *Karesi Municipality Culture and Art House*: Location

- 17 Aralık 2020 Perşembe Saat 20.30 / *Thursday, December 17, 2020 at 20.30*: Time

In TW-2013, the *Ali Sami Yen Turk Telekom Arena Stadium* is referred to as "NEF Stadium", "Ali Sami Yen Stadium", or *Ali Sami Yen Sports Complex Nef Stadium* in Wikipedia pages; hence, the extra information our model received was not sufficient to capture all the words in this entity. On the other hand, the CRF layer was able to understand the relation of the word *stadium* and previous tokens.

In the TW-SUNLP dataset, there is no entry for *Paris Saint-Germain Fan Token* in Wikipedia; therefore, our pipeline cannot provide information about this entity. Also, we cannot find a page for the *Karesi Belediyesi Kültür ve Sanat Evi* in Wikipedia as well. But, including different knowledge bases other than Wikipedia will increase the chance of catching these kinds of entities.

Although extending the vanilla BERT model with a CRF layer or including external knowledge yielded better performances, the time to train and evaluate these models increased significantly. To investigate how efficient these approaches are, we compared the efficiency of the models by examining their training and testing times in Table 6.3 for each dataset. The duration is significantly shorter for informal datasets due to the small number of sentences and the small number of words in each sentence. The training time of the BiLSTM-CRF model is quite short compared to other complex models since it has a simpler structure. Considering that it performs close to the BERT model for formal datasets, it falls almost in the middle of the effectiveness-efficiency threshold.

Implementing a CRF layer significantly increased the required amount of time for training compared to the BERT with a softmax layer. Despite providing a significant increase in scores, the computation time has increased approximately four times for some of the datasets compared to the feed-forward BERT models. The addition of the BiLSTM layer to the BERT-CRF model did not cause a significant time increase for the same number of epochs. However, there was no significant improvement in performance either.

Table 6.3 The training and testing time of neural models and $\text{EL}_{\text{Semantic}}$ for all datasets. The upper row in each dataset shows the training time. The lower row in each dataset shows the testing time.

Dataset	BiLSTM-CRF	BERTurk	+CRF	+BiLSTM-CRF	$\text{EL}_{\text{Semantic}}$
Milliyet	0:49:56	2:55:10	4:26:45	4:34:32	3:09:37
	0:00:23	0:01:17	0:00:54	0:00:59	0:02:44
WikiANN	0:15:22	0:26:43	3:49:52	3:50:46	2:14:23
	0:00:34	0:01:17	0:01:54	0:02:11	0:06:28
TW-2013	0:03:52	0:05:42	0:20:09	0:22:32	0:20:49
	0:00:09	0:00:27	0:00:29	0:00:30	0:00:41
IWT	0:03:40	0:05:49	0:20:13	0:22:20	0:20:27
	0:00:09	0:00:29	0:00:29	0:00:30	0:00:57
TW-SUNLP	0:06:21	0:08:34	0:21:32	0:23:21	0:18:20
	0:00:13	0:00:31	0:00:34	0:00:36	0:01:12
MultiCoNER	0:16:45	0:25:33	1:52:05	2:18:05	1:44:56
	0:04:34	0:11:43	0:20:11	0:22:07	0:48:42

Naturally, since we raised the input length to its maximum value, the training time has increased when we compare $\mathrm{EL}_{\mathrm{Semantic}}$ to the BERTurk model. However, the amount of increase in required time has approached the same time as adding the CRF layer for several datasets. One of the factors affecting this is that we have significantly reduced the batch size due to the small amount of memory in our GPU. Also, the input size for all samples was remarkably increased. Another drawback of our approach is that while the time spent on training is low compared to adding a CRF layer, testing requires more time for $\mathrm{EL}_{\mathrm{Semantic}}$. Moreover, the amount of time spent on our pipeline is not limited to these values.

Table 6.4 The time spends on the Candidate Generation, Mention Detection, and Re-ranking parts in our pipeline.

Dataset	Searching	Matching	Re-ranking
Milliyet	0:35:06	0:04:43	07:04:48
WikiANN	0:27:39	0:03:46	06:54:12
TW-2013	0:05:19	0:01:29	01:15:23
IWT	0:05:28	0:01:21	01:21:02
TW-SUNLP	0:05:44	0:01:35	01:42:14
MultiCoNER	0:24:53	0:03:30	05:48:10

In Table 6.4, we present the time spent preparing the external context. Indexing for more than 850,000 Wikipedia pages on ElasticSearch took only about 3 minutes. Matching the possible entities from the input samples to the Wikipedia pages also requires a small amount of time. However, searching input samples and especially re-ranking the documents takes an excessive amount of time. In particular, the re-ranking of retrieved documents is a bottleneck in our pipeline, which makes this method inefficient. In this study, however, we have implemented a simple approach that can be improved in several ways. For instance, indexing documents directly with contextual representation might increase both efficiency and effectiveness.

7. CONCLUSION

In this thesis, we address the difficulties of detecting complex entities and the lack of context in noisy texts for the Named Entity Recognition task. To alleviate these problems, we introduced two approaches that enhance the state-of-the-art transformer-based models by integrating external context from a knowledge base. Besides, as part of this research, we introduced a new Twitter dataset to eliminate the inadequacy of the social media datasets in Turkish. Furthermore, we explored the impact of implementing extra layers, such as CRF and BiLSTM, to the transformer-based models and compared the effectiveness of external knowledge with extra layers in MultiCoNER and Twitter datasets full of noisy, short, and contains complex entities. It should be mentioned that most of the research covered in this study was published at the 16th International Workshop on Semantic Evaluation (Çarık et al., 2022) and the Language Resources and Evaluation Conference (Çarık & Yeniterzi, 2022).

In order to incorporate external knowledge, we proposed two straightforward approaches that utilized Wikipedia pages as additional context. After retrieving the related pages with a search engine that used our input samples as a query, first, we tried to detect and classify the possible named entities by finding exact matches in the retrieved documents within the input samples. In our second approach, we improved the previous method by re-ranking the retrieved documents according to their contextual similarity. Next, we concatenated the five most relevant documents with input samples and fed them into the BERT model to predict named entity classes. The second method improved over the vanilla BERT model in almost all datasets. When we further examine the performance of this method, our findings showed that external knowledge is beneficial for detecting long entities as well as entities belonging to the complex named entity classes, even though the current implementation is not efficient.

In conclusion, we revisit our research questions to answer based on our findings:

RQ1: *Would implementing extra layers on top of the transformer-based models* *outperform only using the softmax function?*

Adding a CRF layer to the transformer-based models achieved state-of-the-art results in all datasets, except for the MultiCoNER dataset. The lack of context in the examples in the MultiCoNER caused the models to fail without additional information. Also, the BERT-CRF model suffered more in classes with complex and uncertain entities. On the other hand, the BiLSTM layer has not achieved significant success above the BERT-CRF model.

RQ2: *Would integrating an external context from a knowledge base like Wikipedia into transformer-based models improve results for NER in Turkish?*

Our experiments showed that external context helps transformer-based models to improve their performance in various domains, especially for the texts that are short in context and contains complex and ambiguous named entity classes. Highlighting semantically relevant content has been further enhanced as it provides additional information even if the system cannot find most of the entities in the Mention Detection section.

We answer the sub-questions by discussing our results in more detail:

RQ2-1: *In what conditions would utilizing the Wikipedia pages of detected possible named entities as external knowledge improve the NER?*

Using pages of possible entities as external knowledge improved the results in the MultiCoNER dataset that contains many short sentences with insufficient context. Also, our $\mathrm{EL}_{\mathrm{BERT}}$ and $\mathrm{EL}_{\mathrm{MultiBERT}}$ pipelines effectively detect complex and ambiguous entities since they showed the most remarkable improvements in these classes, like Product and Creative Work in MultiCoNer.

RQ2-2: *Would highlighting semantically similar texts to input samples be more effective than using only pages of detected possible named entities?*

Emphasizing semantically closer pages achieved better results than utilizing only detected named entities and even outperformed the BERTurk model in all noisy datasets. However, $\mathrm{EL}_{\mathrm{Semantic}}$ is not an efficient pipeline as re-ranking is a time-consuming process, and testing time increases significantly.

7.1 Future Work

As seen from the results, the inclusion of external knowledge is a promising method for the Named Entity Recognition task. However, the improvements we achieved were modest due to the simplicity of our approach. For instance, using the exact match method in the search engine to find relevant documents ignores synonyms and semantically related phrases. We are constrained by the capacity of the initially retrieved documents, even if we subsequently re-ranked based on contextual similarity. Also, the re-ranking process takes a long time; thus, it makes our pipeline ineffective. In order to alleviate these issues, neural search engines that index and search documents over contextual representations might be applied.

One of the problems that hinder the performance of models in informal datasets and degrades the quality of retrieved documents is the noisy nature of social media data. The quality of the retrieved documents and the re-ranking process for these data can be improved by applying various approximation methods.

Moreover, the performance of the BERT model was significantly improved by adding a CRF layer on top of it. Scores of the model that leverages external knowledge might be boosted even more by implementing a CRF layer.

Finally, to increase the relatedness of the retrieved content and cover more named entities in the search results, we might expand the utilized knowledge bases and experiment with alternative search engines, like Google Search. Furthermore, to detect foreign named entities that are particularly common in informal datasets, multilingual Wikipedia pages can also be indexed.

BIBLIOGRAPHY

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, (pp. 1638–1649).

Akkaya, E. K. & Can, B. (2021). Transfer learning for turkish named entity recognition on noisy text. *Natural Language Engineering*, *27*(1), 35–64.

Alecakir, H., Bölücü, N., & Can, B. (2022). Turkishdelightnlp: A neural turkish nlp toolkit. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, (pp. 17–26).

Alexis, C., Kartikay, K., Naman, G., Vishrav, C., Guillaume, W., Francisco, G., Edouard, G., Myle, O., Luke, Z., & Veselin, S. (2019). Unsupervised cross-lingual representation learning at scale. *CoRR*, *abs/1911.02116*.

Aras, G., Makaroğlu, D., Demir, S., & Cakir, A. (2021). An evaluation of recent neural sequence tagging models in turkish named entity recognition. *Expert Systems with Applications*, *182*, 115049.

Arkhipov, M., Trofimova, M., Kuratov, Y., & Sorokin, A. (2019). Tuning multilingual transformers for named entity recognition on slavic languages. In *Proc. of 7th Workshop on Balto-Slavic Natural Language Processing (BSNLP'19)*, (pp. 89–93).

Augenstein, I., Derczynski, L., & Bontcheva, K. (2017). Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, *44*, 61–83.

Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, (pp. 194–201).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.

Çarık, B., Beyhan, F., & Yeniterzi, R. (2022). SU-NLP at SemEval-2022 task 11: Complex named entity recognition with entity linking. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, (pp. 1648–1653)., Seattle, United States. Association for Computational Linguistics.

Çarık, B. & Yeniterzi, R. (2022). A twitter corpus for named entity recognition in turkish. In *Proceedings of the Language Resources and Evaluation Conference*, (pp. 4546–4551)., Marseille, France. European Language Resources Association.

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). IEEE.

Çetindağ, C., Yazıcıoğlu, B., & Koç, A. (2022). Named-entity recognition in turkish legal texts.

Chen, B., Ma, J.-Y., Qi, J., Guo, W., Ling, Z.-H., & Liu, Q. (2022). USTC-

NELSLIP at SemEval-2022 task 11: Gazetteer-adapted integration network for multilingual complex named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, (pp. 1613–1622)., Seattle, United States. Association for Computational Linguistics.

Chinchor, N. A. (1998). Overview of MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.*

Chiu, J. P. & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, *4*, 357–370.

Collobert, R. & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine learning*, (pp. 160–167).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(ARTICLE), 2493–2537.

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

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

Ding, R., Xie, P., Zhang, X., Lu, W., Li, L., & Si, L. (2019). A neural multi-digraph model for chinese ner with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (pp. 1462–1467).

Eddy, S. R. (1996). Hidden markov models. *Current opinion in structural biology*, *6*(3), 361–365.

Eken, B. & Tantuğ, A. (2015). Recognizing named entities in turkish tweets. *Computer Science & Information Technology*, *5*, 155–162.

Fetahu, B., Fang, A., Rokhlenko, O., & Malmasi, S. (2021). Gazetteer enhanced named entity recognition for code-mixed web queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 1677–1681).

Fetahu, B., Fang, A., Rokhlenko, O., & Malmasi, S. (2022). Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 2777–2790).

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

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

Grishman, R. & Sundheim, B. M. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics.*

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

Güngör, O. & Yıldız, E. (2017). Linguistic features in turkish word representations. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, (pp. 1–4). IEEE.

Güngör, O., Üsküdarlı, S., & Güngör, T. (2018). Recurrent neural networks for turkish named entity recognition. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, (pp. 1–4).

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

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

Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., & Wilks, Y. (1998). University of sheffield: Description of the lasie-ii system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Jiang, Z.-H., Yu, W., Zhou, D., Chen, Y., Feng, J., & Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, *33*, 12837–12848.

Krishnan, V. & Manning, C. D. (2006). An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, (pp. 1121–1128).

Krupka, G. & IsoQuest, K. (2005). Description of the nerowl extractor system as used for muc-7. In *Proc. 7th Message Understanding Conference*, (pp. 21–28).

Küçük, D. & Can, F. (2019). A tweet dataset annotated for named entity recognition and stance detection.

Küçük, D. et al. (2009). Named entity recognition experiments on turkish texts. In *International Conference on Flexible Query Answering Systems*, (pp. 524–535). Springer.

Küçük, D., Jacquet, G., & Steinberger, R. (2014). Named entity recognition on turkish tweets. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, (pp. 450–454).

Küçük, D., Küçük, D., & Arıcı, N. (2016). A named entity recognition dataset for turkish. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, (pp. 329–332). IEEE.

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)*, (pp. 71–78)., Gothenburg, Sweden. Association for Computational Linguistics.

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

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

Kuru, O., Can, O. A., & Yuret, D. (2016). Charner: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, (pp. 911–921).

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, (pp. 282–289)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, (pp. 260–270).

Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, *34*(1), 50–70.

Lin, H., Lu, Y., Han, X., Sun, L., Dong, B., & Jiang, S. (2019). Gazetteer-enhanced attentive neural networks for named entity recognition. 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)*, (pp. 6232–6237).

Liu, T., Yao, J.-G., & Lin, C.-Y. (2019). Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (pp. 5301–5307).

Liu, X., Zhang, S., Wei, F., & Zhou, M. (2011). Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (pp. 359–367).

Loshchilov, I. & Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ma, L., Jian, X., & Li, X. (2022). Pai at semeval-2022 task 11: Name entity recognition with contextualized entity representations and robust loss functions. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, (pp. 1665–1670).

Malmasi, S., Fang, A., Fetahu, B., Kar, S., & Rokhlenko, O. (2022a). MultiCoNER: A large-scale multilingual dataset for complex named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, (pp. 3798–3809)., Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Malmasi, S., Fang, A., Fetahu, B., Kar, S., & Rokhlenko, O. (2022b). Semeval-2022 task 11: Multilingual complex named entity recognition (multiconer). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, (pp. 1412–1437).

Manchanda, P., Fersini, E., & Palmonari, M. (2015). Leveraging entity linking to enhance entity recognition in microblogs. In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, volume 1, (pp. 147–155). IEEE.

Mayhew, S., Tsygankova, T., & Roth, D. (2019). ner and pos when nothing is capitalized. 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)*, (pp. 6256–6261).

McCallum, A., Freitag, D., & Pereira, F. C. (2000). Maximum entropy markov models for information extraction and segmentation. In *ICML*, volume 17, (pp. 591–598).

McCallum, A. & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.

McNamee, P. & Mayfield, J. (2002). Entity extraction without language-specific resources. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Meng, T., Fang, A., Rokhlenko, O., & Malmasi, S. (2021). Gemnet: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 1499–1512).

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR*.

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

Oflazer, K., Göçmen, E., & Bozşahin, C. (1994). An outline of turkish morphology.

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)*, (pp. 549–555)., Portorož, Slovenia. European Language Resources Association (ELRA).

Oluk, A. & Özgur, H. (2020). Turkish language models. `https://github.com/loodos/turkish-language-models`.

Onal, K. D. & Karagoz, P. (2015). Named entity recognition from scratch on social media. In *Proceedings of 6th International Workshop on Mining Ubiquitous and Social Environments (MUSE), co-located with the ECML PKDD*, volume 104.

Ozcelik, O. & Toraman, C. (2022). Named entity recognition in turkish: A comparative study with detailed error analysis. *Information Processing & Management*, *59*(6), 103065.

Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., & Ji, H. (2017). Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (pp. 1946–1958).

Passos, A., Kumar, V., & McCallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, (pp. 78–86).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532–1543).

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations. 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)*, (pp. 43–54)., Hong Kong, China. Association for Computational Linguistics.

Rabiner, L. & Juang, B. (1986). An introduction to hidden markov models. *IEE ASSP Magazine*, *3*(1), 4–16.

Ratinov, L. & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, (pp. 147–155)., USA. Association for Computational Linguistics.

Ritter, A., Clark, S., Mausam, M., & Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, (pp. 1524–1534).

Safaya, A., Kurtuluş, E., Goktogan, A., & Yuret, D. (2022). Mukayese: Turkish NLP strikes back. In *Findings of the Association for Computational Linguistics: ACL 2022*, (pp. 846–863)., Dublin, Ireland. Association for Computational Linguistics.

Sang, E. T. K. & Veenstra, J. (1999). Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, (pp. 173–179).

Şeker, G. A. & Eryiğit, G. (2012). Initial explorations on using crfs for turkish named entity recognition. In *Proceedings of COLING 2012*, (pp. 2459–2474).

Seker, G. A. & Eryigit, G. (2017). Extending a crf-based named entity recognition model for turkish well formed text and user generated content. *Semantic Web*, *8*(5), 625–642.

Song, C. H., Lawrie, D., Finin, T., & Mayfield, J. (2020). Improving neural named entity recognition with gazetteers.

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

Stefan, S. (2020). Berturk - bert models for turkish.

Takeuchi, K. & Collier, N. (2002). Use of support vector machines in extended named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

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

Tür, G., Hakkani-Tür, D., & Oflazer, K. (2003). A statistical information extraction system for turkish. *Natural Language Engineering*, *9*(2), 181–210.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., & Tu, K. (2021). Improving named entity recognition by external context retrieving and cooperative learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (pp. 1800–1812).

Wang, X., Shen, Y., Cai, J., Wang, T., Wang, X., Xie, P., Huang, F., Lu, W., Zhuang, Y., Tu, K., Lu, W., & Jiang, Y. (2022). DAMO-NLP at SemEval-2022 task 11: A knowledge-based system for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, (pp. 1457–1468)., Seattle, United States. Association for Computational Linguistics.

Wang, Z., Qu, Y., Chen, L., Shen, J., Zhang, W., Zhang, S., Gao, Y., Gu, G., Chen, K., & Yu, Y. (2018). Label-aware double transfer learning for cross-specialty medical named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (pp. 1–15)., New Orleans, Louisiana. Association for Computational Linguistics.

Yamada, I., Asai, A., Shindo, H., Takeda, H., & Matsumoto, Y. (2020). Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 6442–6454).

Yamada, I., Takeda, H., & Takefuji, Y. (2015). Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text*, (pp. 136–140).

Yeniterzi, R. (2011). Exploiting morphology in turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session*, (pp. 105–110).

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhang, Y. & Yang, J. (2018). Chinese ner using lattice lstm. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (pp. 1554–1564).

APPENDIX A

Table A.1 Data distributions for Milliyet and WikiANN.

	Milliyet			**WikiANN**		
	Train	Validation	Test	Train	Validation	Test
Person	13,290	1,400	1,603	8,833	4,374	4,519
Location	8,821	942	1,126	9,679	5,014	4,914
Organization	8,316	842	873	8,833	4,374	4,519
Sentences	22,338	2,482	2,751	20,000	10,000	10,000
Tokens	419,996	45,532	49,600	149,786	75,930	75,731
NE	30,427	3,184	3,602	26,482	13,517	13,587
Unique NE	6,927	1,523	1,408	16,584	9,485	9,551

Table A.2 Data distributions for TW-2013 and IWT.

	TW-2013			**IWT**		
	Train	Validation	Test	Train	Validation	Test
Person	558	64	75	327	30	23
Location	145	20	19	212	24	24
Organization	361	29	34	340	29	32
Money	6	2	4	41	2	2
Date	45	4	7	50	2	7
Time	16	2	2	7	0	2
Percent	2	0	1	5	2	1
Sentences	4,081	454	504	4,058	451	502
Tokens	37,708	4,318	4,722	38,406	4,286	4,667
NE	1,136	121	142	982	89	91
Unique NE	755	109	122	567	73	78

Table A.3 TW-SUNLP and MultiCoNER data distributions in detail

	TW-SUNLP			**MultiCoNER**		
	Train	Validation	Test	Train	Validation	Test
Person	3,984	768	830	4,414	231	26,876
Location	866	165	194	5,804	351	34,609
Organization	2,056	464	457	-	-	-
Money	124	29	17	-	-	-
Time	414	114	105	-	-	-
TV-Show	204	52	48	-	-	-
Product	261	74	45	3,184	158	21,388
Group	-	-	-	3,568	167	21,951
Corporation	-	-	-	2,761	148	21,137
Creative Work	-	-	-	3,574	190	23,408
Sentences	3,500	750	750	15,300	800	136,935
Tokens	87,704	18,736	18,702	218,399	11,417	723,226
NE	7,909	1,666	1,696	23,305	1,245	149,369
Unique NE	4,745	1,264	1,307	11,601	1,037	89,333

Table A.4 Example sentences from six datasets used in this study. The second column presents the original words and corresponding labels based on the IOB2 tagging scheme for each dataset. The third column is the English translation of the sentence.

Dataset	Sentence							Translation
Milliyet	Sadece	Sergen	Kartal'ı	ürküten	bir	oyuncu	oldu	Only Sergen was a player that frightened Kartal.
	O	B-PER	B-ORG	O	O	O	O	
WikiANN	Ünlü	yönetmen	Dupetron	Tartas	Landes	Fransa	doğumludur	Famous director Dupetron Tartas was born in Landes, France
	O	O	B-PER	B-LOC	B-LOC	B-LOC	O	
TW-2013	Ben	ıngıltereye	gıttıgımde	harry	nın	evıne	gıdıcem	When I go to England, I will go to Harry's house.
	O	B-LOC	O	B-PER	O	O	O	
IWT	BBG	evine	döndü	.				BBG returned its homeland.
	B-ORG	O	O	O				
TW-SUNLP	Herife	bak	sanki	Osm'de	coinsle	lig	kuruyor	Look at the guy, it's like he's building a league with coins in Osm
	O	O	O	B-ORG	O	O	O	
MultiCoNER	dave	clark	nedir					What's dave clark
	B-PER	I-PER	O					

Table A.5 The performance of BiLSTM-CRF model with different word embeddings in validation and test sets of formal and informal datasets.

	Word	**Validation**			**Test**		
	Embedding	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Milliyet	FT-FB	91.36±.00	91.55±.00	91.44±.00	91.60±.01	90.20±.00	90.89±.01
	FT-SUNLP	84.11±.01	85.08±.01	84.52±.01	84.95±.01	84.50±.01	84.69±.01
	W2V-Gungor	93.43±.00	94.68±.00	94.05±.00	93.42±.00	93.84±.00	93.63±.00
	W2V-SUNLP	85.27±.01	81.73±.01	83.45±.01	84.51±.01	79.90±.01	82.10±.01
	GloVe	90.10±.01	90.03±.01	90.06±.01	90.73±.00	89.07±.00	89.89±.00
WikiANN	FT-FB	88.40±.01	86.15±.01	87.23±.01	88.02±.01	85.24±.00	86.57±.00
	FT-SUNLP	81.90±.01	80.34±.01	80.79±.01	81.48±.00	79.46±.01	80.13±.00
	W2V-Gungor	89.93±.00	88.50±.01	89.19±.00	89.83±.00	87.91±.01	88.83±.00
	W2V-SUNLP	75.41±.00	68.39±.00	71.65±.00	75.65±.00	67.67±.00	71.34±.00
	GloVe	84.88±.01	82.34±.01	83.54±.01	84.58±.01	81.41±.01	82.89±.01
TW-2013	FT-FB	66.46±.03	56.69±.02	60.15±.02	69.73±.03	56.62±.02	62.22±.02
	FT-SUNLP	63.87±.03	32.23±.04	41.77±.03	71.23±.05	39.01±.02	48.85±.02
	W2V-Gungor	69.81±.04	45.95±.02	54.99±.02	74.43±.02	52.25±.02	60.83±.01
	W2V-SUNLP	65.49±.01	57.36±.01	60.81±.01	66.96±.03	57.75±.01	61.70±.02
	GloVe	68.73±.04	57.85±.03	62.11±.02	67.69±.02	55.07±.04	59.99±.03
IWT	FT-FB	76.08±.02	71.69±.04	73.60±.03	80.20±.02	70.55±.03	74.47±.03
	FT-SUNLP	73.10±.03	56.63±.01	63.51±.01	68.11±.04	41.54±.03	51.02±.03
	W2V-Gungor	81.23±.01	77.30±.05	78.98±.03	77.86±.04	70.99±.02	74.04±.02
	W2V-SUNLP	71.95±.02	73.93±.02	72.74±.02	70.66±.02	69.23±.01	69.47±.01
	GloVe	72.99±.03	71.69±.03	71.91±.03	74.15±.03	75.16±.01	74.17±.02
TW-SUNLP	FT-FB	76.49±.02	64.32±.01	68.93±.01	76.32±.02	63.55±.01	68.60±.01
	FT-SUNLP	65.50±.02	50.78±.01	55.56±.01	66.28±.02	51.98±.01	57.19±.01
	W2V-Gungor	76.54±.01	65.04±.01	69.43±.01	75.27±.01	64.79±.00	69.27±.00
	W2V-SUNLP	73.78±.00	63.89±.00	67.85±.00	72.46±.00	61.12±.01	65.72±.01
	GloVe	76.42±.00	65.63±.00	69.85±.00	76.35±.01	64.91±.01	69.51±.01
MultiCoNER	FT-FB	79.73±.01	78.44±.01	78.95±.00	59.11±.01	44.10±.01	49.84±.01
	FT-SUNLP	74.75±.02	69.61±.02	71.71±.01	54.51±.01	38.06±.01	43.19±.01
	W2V-Gungor	76.39±.00	72.79±.01	74.44±.01	51.41±.01	33.27±.01	39.33±.01
	W2V-SUNLP	69.36±.01	65.78±.01	67.05±.01	48.08±.00	26.88±.01	34.11±.01
	GloVe	77.09±.01	76.67±.01	76.72±.00	55.68±.01	39.65±.01	45.97±.01

Table A.6 The performance of transformer-based models in validation and test sets of **Milliyet** dataset

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
BERT	mBERT	90.77±.01	92.00±.01	91.36±.01	87.64±.01	88.28±.00	87.93±.01
	XLM-R	90.88±.02	93.27±.01	92.04±.01	88.94±.01	91.06±.01	89.96±.01
	BERTurk	95.00±.00	96.62±.00	95.80±.00	94.84±.01	95.89±.01	95.35±.01
	BERT$_{loodos}$	94.23±.01	96.04±.00	95.12±.00	93.64±.01	94.74±.01	94.17±.01
	ConvBERTurk	94.69±.01	96.39±.00	95.52±.00	95.12±.01	95.88±.01	95.49±.00
+CRF	BERTurk	95.61±.00	96.95±.00	96.27±.00	95.53±.00	95.91±.00	95.72±.00
	BERT$_{loodos}$	95.70±.00	97.12±.00	96.40±.00	95.66±.00	96.00±.00	95.82±.00
	ConvBERTurk	95.54±.00	96.93±.00	96.23±.00	96.00±.00	95.87±.00	95.94±.00
+BiLSTM-CRF	BERTurk	95.54±.00	96.70±.00	96.11±.00	95.76±.00	95.97±.00	95.87±.00
	BERT$_{loodos}$	95.36±.00	96.65±.00	96.00±.00	95.26±.00	95.67±.00	95.46±.00
	ConvBERTurk	95.42±.00	96.79±.00	96.10±.00	95.50±.00	96.01±.00	95.76±.00

Table A.7 The performance of transformer-based models in validation and test sets of **WikiANN** dataset

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
BERT	mBERT	90.94±.00	92.65±.00	91.78±.00	90.84±.00	92.24±.00	91.52±.00
	XLM-R	89.61±.00	91.06±.00	90.32±.00	89.43±.00	90.83±.00	90.12±.00
	BERTurk	90.72±.01	92.60±.01	91.62±.01	90.57±.01	92.18±.01	91.34±.01
	BERT_{loodos}	90.80±.00	92.37±.00	91.57±.00	90.86±.00	92.08±.01	91.45±.00
	ConvBERTurk	91.76±.00	93.18±.00	92.46±.00	91.57±.01	92.57±.00	92.06±.00
+CRF	BERTurk	93.26±.00	94.34±.00	93.80±.00	93.33±.00	94.08±.00	93.70±.00
	BERT_{loodos}	92.12±.01	93.61±.00	92.86±.01	92.23±.01	93.32±.00	92.77±.01
	ConvBERTurk	93.12±.00	94.34±.00	93.72±.00	93.04±.00	93.76±.00	93.40±.00
+BiLSTM-CRF	BERTurk	93.06±.00	94.16±.00	93.60±.00	93.11±.00	93.97±.00	93.54±.00
	BERT_{loodos}	92.42±.00	93.72±.00	93.07±.00	92.47±.00	93.33±.00	92.90±.00
	ConvBERTurk	92.71±.00	93.98±.00	93.34±.00	92.75±.00	93.57±.00	93.16±.00

Table A.8 The performance of transformer-based models in validation and test sets of **TW-2013** dataset

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
BERT	mBERT	68.17±.07	57.69±.02	61.02±.03	66.33±.05	55.77±.03	59.89±.04
	XLM-R	66.12±.02	66.94±.03	66.16±.02	65.86±.04	64.65±.01	64.58±.03
	BERTurk	68.93±.05	71.24±.04	69.55±.04	67.74±.07	70.28±.03	68.38±.04
	BERT_{loodos}	68.19±.04	68.26±.03	67.81±.01	67.50±.04	67.32±.04	66.75±.01
	ConvBERTurk	67.49±.05	73.55±.02	69.78±.02	68.13±.02	71.55±.06	69.41±.03
+CRF	BERTurk	70.41±.02	71.57±.02	70.82±.02	74.20±.02	74.23±.02	73.52±.02
	BERT_{loodos}	64.05±.03	68.10±.02	65.85±.02	69.13±.02	70.28±.01	69.38±.02
	ConvBERTurk	66.05±.01	71.90±.02	68.58±.02	71.75±.02	70.70±.02	70.96±.02
+BiLSTM-CRF	BERTurk	58.08±.03	60.99±.05	59.35±.04	61.30±.07	67.75±.06	63.95±.06
	BERT_{loodos}	63.71±.04	65.29±.02	64.33±.03	64.12±.02	63.80±.03	63.67±.03
	ConvBERTurk	64.11±.03	67.93±.02	65.58±.02	63.10±.03	64.79±.03	63.19±.03

Table A.9 The performance of transformer-based models in validation and test sets of **IWT** dataset

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
BERT	mBERT	77.16±.05	81.12±.02	78.73±.02	74.18±.06	71.43±.03	72.00±.04
	XLM-R	77.03±.04	79.33±.09	77.82±.06	70.91±.04	70.11±.09	69.65±.07
	BERTurk	83.64±.02	93.03±.03	87.75±.02	83.60±.03	88.57±.01	85.58±.02
	BERT_{loodos}	83.75±.02	91.69±.02	87.14±.01	81.66±.04	86.15±.03	83.20±.03
	ConvBERTurk	84.66±.02	91.91±.04	87.75±.02	82.45±.02	85.49±.06	83.26±.04
+CRF	BERTurk	85.65±.01	93.48±.02	89.11±.01	85.78±.03	90.33±.02	87.49±.02
	BERT_{loodos}	86.29±.01	92.36±.01	88.95±.01	85.25±.03	86.15±.04	85.27±.03
	ConvBERTurk	83.98±.02	90.11±.02	86.57±.02	83.79±.02	83.08±.02	82.86±.01
+BiLSTM-CRF	BERTurk	72.91±.10	82.92±.07	76.89±.09	74.77±.09	82.20±.06	77.63±.08
	BERT_{loodos}	79.80±.03	86.74±.02	82.67±.02	76.49±.04	79.34±.05	77.26±.05
	ConvBERTurk	78.02±.02	88.31±.03	82.35±.02	74.59±.03	80.22±.03	76.94±.03

Table A.10 The performance of transformer-based models in validation and test sets of **TW-SUNLP** dataset

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
BERT	mBERT	77.18±.01	76.55±.01	76.78±.00	76.04±.01	75.24±.01	75.40±.00
	XLM-R	78.47±.02	81.60±.01	79.80±.01	76.68±.01	79.47±.01	77.81±.01
	BERTurk	84.80±.01	86.90±.01	85.74±.00	83.05±.01	85.37±.01	84.01±.00
	BERT$_{loodos}$	84.77±.00	85.97±.01	85.27±.00	84.14±.01	85.27±.01	84.59±.01
	ConvBERTurk	84.57±.01	87.31±.00	85.81±.01	83.98±.01	86.49±.00	85.04±.01
+CRF	BERTurk	85.31±.01	88.18±.01	86.68±.01	84.44±.00	87.28±.00	85.68±.00
	BERT$_{loodos}$	84.78±.00	87.07±.01	85.86±.00	84.22±.01	86.66±.01	85.24±.01
	ConvBERTurk	85.87±.01	88.93±.01	87.32±.01	85.23±.00	87.89±.00	86.36±.00
+BiLSTM-CRF	BERTurk	84.18±.01	86.33±.01	85.17±.01	83.56±.01	85.41±.01	84.29±.01
	BERT$_{loodos}$	83.84±.01	85.31±.00	84.52±.01	82.93±.01	85.38±.00	83.99±.01
	ConvBERTurk	84.23±.01	86.72±.01	85.36±.01	84.85±.00	87.00±.01	85.74±.00

Table A.11 The performance of transformer-based models in validation and test sets of **MultiCoNER** dataset

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
BERT	mBERT	78.48±.02	81.24±.02	79.69±.01	44.66±.02	47.02±.01	44.87±.01
	XLM-R	78.13±.02	80.50±.02	79.15±.01	48.21±.02	51.29±.01	48.85±.01
	BERTurk	80.36±.01	85.33±.00	82.70±.00	48.26±.01	53.07±.01	49.95±.01
	BERT$_{loodos}$	79.87±.01	84.77±.01	82.16±.00	46.60±.01	52.58±.01	48.83±.01
	ConvBERTurk	82.78±.01	85.16±.01	83.89±.00	52.71±.01	56.81±.01	54.21±.01
+CRF	BERTurk	81.54±.01	85.85±.00	83.59±.01	50.16±.00	55.71±.01	52.31±.00
	BERT$_{loodos}$	81.64±.01	85.06±.01	83.27±.01	50.12±.01	54.82±.01	51.98±.00
	ConvBERTurk	82.34±.01	86.39±.00	84.28±.01	54.29±.01	59.99±.00	56.64±.01
+BiLSTM-CRF	BERTurk	80.66±.00	85.59±.01	83.01±.01	49.22±.01	54.38±.01	51.33±.01
	BERT$_{loodos}$	79.94±.01	84.95±.01	82.30±.01	48.51±.01	53.34±.00	50.55±.00
	ConvBERTurk	81.88±.00	86.55±.00	84.12±.00	55.21±.00	59.52±.00	57.03±.00

Table A.12 The performances of knowledge-base approaches in validation and test sets for six datasets.

		Validation			**Test**		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
Milliyet	EL$_{\text{BERT}}$	52.58±.00	69.20±.00	59.71±.00	47.63±.00	65.14±.01	54.95±.00
	EL$_{\text{MultiBERT}}$	52.34±.01	69.53±.00	59.69±.00	46.91±.01	65.84±.01	54.72±.00
	EL$_{\text{Semantic}}$	94.07±.01	94.82±.01	94.43±.01	94.13±.01	93.98±.01	94.04±.01
WikiANN	EL$_{\text{BERT}}$	56.91±.00	71.72±.00	63.04±.00	56.17±.00	70.90±.00	62.33±.00
	EL$_{\text{MultiBERT}}$	56.64±.00	71.74±.00	62.91±.00	55.79±.00	70.82±.00	62.09±.00
	EL$_{\text{Semantic}}$	91.97±.00	93.35±.01	92.65±.00	92.13±.01	93.53±.00	92.82±.00
TW-2013	EL$_{\text{BERT}}$	42.31±.09	64.88±.08	50.67±.09	29.86±.09	46.63±.04	35.67±.08
	EL$_{\text{MultiBERT}}$	45.79±.02	68.33±.05	54.07±.01	30.28±.06	44.95±.06	35.78±.06
	EL$_{\text{Semantic}}$	70.63±.05	73.88±.03	71.72±.02	72.08±.04	76.20±.04	73.49±.01
IWT	EL$_{\text{BERT}}$	60.45±.03	63.16±.02	60.94±.02	68.79±.03	69.28±.02	67.86±.02
	EL$_{\text{MultiBERT}}$	61.80±.01	65.46±.04	63.26±.02	69.23±.05	69.88±.03	68.42±.03
	EL$_{\text{Semantic}}$	82.01±.03	91.91±.02	86.34±.02	85.19±.02	89.45±.03	86.83±.01
TW-SUNLP	EL$_{\text{BERT}}$	45.83±.00	64.74±.01	53.41±.00	45.40±.00	65.76±.01	53.42±.00
	EL$_{\text{MultiBERT}}$	44.93±.01	65.99±.01	53.06±.00	44.85±.00	66.72±.00	53.32±.00
	EL$_{\text{Semantic}}$	85.07±.01	87.56±.00	86.22±.01	83.78±.01	86.12±.00	84.77±.01
MultiCoNER	EL$_{\text{BERT}}$	80.06±.02	82.24±.02	81.00±.00	63.80±.04	73.92±.01	68.07±.03
	EL$_{\text{MultiBERT}}$	79.84±.02	84.29±.01	81.92±.01	63.36±.01	77.17±.02	69.32±.01
	EL$_{\text{Semantic}}$	84.66±.01	86.73±.01	85.64±.00	67.92±.01	71.92±.02	69.67±.01

Table A.13 Number of matched possible entities with empty pages

	Empty Pages		
	Train	Validation	Test
Milliyet	3,839	420	371
WikiANN	845	442	456
TW-2013	169	27	24
TW-SUNLP	437	79	98
IWT	379	46	42
MultiCoNER	2,231	125	5,209

Table A.14 Support values for each dataset

Named Entity	Milliyet	WikiANN	TW-2013	IWT	TW-SUNLP	MultiCoNER
Person	1,603	4,519	75	23	830	26,876
Location	1,126	4,914	19	24	194	34,609
Organization	873	4,154	34	32	457	-
Money	-	-	4	2	17	-
Time	-	-	2	2	105	-
Date	-	-	7	7	-	-
Percent	-	-	1	1	-	-
Product	-	-	-	-	45	21,388
TV-Show	-	-	-	-	48	-
Corporation	-	-	-	-	-	21,137
Group	-	-	-	-	-	21,951
Creative Work	-	-	-	-	-	23,408
Total	3,602	13,587	142	91	1,696	149,369