Named Entity Recognition Using Bidirectional Encoder Representations from Transformers

Aslı Şeyda Özdemir, Alaaddin Göktuğ Ayar
Department of Computer Engineering
YIldız Technical University, 34220 Istanbul, Turkey
{l1118501, l1119603}@std.yildiz.edu.tr

***Özetçe* —Varlık İsmi Tanıma, günlük hayatta yaygın olarak kullanılan kişi, yer, kuruluş gibi metin belgelerinden önceden tanımlanmış kategorilerin çıkarılması işlemidir. Varlık İsmi Tanıma bilgi çıkarmanın bir alt kategorisi olarak bilinir ve makine çevirileri ve duygu analizi gibi birçok Doğal Dil İşleme probleminde kullanılır. Teknolojinin gelişmesiyle birlikte Varlık İsmi Tanıma yöntemleri gelişmiştir. Bu makalede Türkçe dili özelinde bir çalışma yapılmıştır. Türkçe, morfolojik özellikleri bakımından karmaşık bir yapıya sahiptir. Türkçe dilinde bu konuda sınırlı sayıda çalışma bulunmaktadır. BERT modelini kullanan Türkçe için akademik çalışmalar neredeyse yok denecek kadar azdır. 2018 yılında piyasaya Google tarafından sürülen BERT modelinin hızı ve modelin doğruluğunun çok yüksek olduğu gözlemlenmiştir. Bu çalışma, modelin Türkçe için sonuçlarını gözlemlemeyi hedefler. Bu makale, BERT modelini kullanarak Varlık İsmi Tanıma hakkında bilgi sağlamayı amaçlamaktadır. Amacımız, bir metinde kişi, yer, kurum, zaman, yüzde, tarih, para kategorilerini doğru bir şekilde belirleyebilen bir sistem geliştirmektir.**

Anahtar Kelimeler—BERT (Bidirectional Encoder Representations from Transformers), Varlık İsmi Tanıma, Doğal Dil İşleme, Bilgi Çıkarımı, Duygu Analizi, Makine Çevirileri.

***Abstract*—Named Entity Recognition is the process of extracting predefined categories from text documents such as person, place, organization, which are commonly used in daily life. It is a sub-category of information extraction and is used in many Natural Language Processing problems such as machine translations and sentiment analysis. With the development of technology, Named Entity Recognition methods have developed. The Turkish language has a complex structure in terms of its morphological features. At the same time, there are a limited number of studies on Named Entity Recognition for Turkish. Academic studies for Turkish using the BERT model are almost non-existent. This article aims to provide information about named entity recognition using the BERT model. Our goal is the implement a BERT model that can accurately determine the expressions of the person, place,institution, time, percentage, date, money categories in a text for the Turkish language.**

Keywords—BERT (Bidirectional Encoder Representations from Transformers), Information Extraction, Named Entity Recognition).

I. Introduction

Named Entity Recognition - NER is an information extraction problem in natural language processing that aims to extract predefined categories such as a person, place, location, time, date and organization from text documents. Although named entity recognition generally consists of these tags, different tags can be used in special studies. It can be used for classifying content for publishers, search engines to provide best searching results, powering content recommendations for some applications, customer support and so on. This technology first came out in 1995 at the MUC-6 conference. In 1995, named entity recognition definition has been made and 3 basic categories created, namely ENAMEX, TIMEX and NUMEX. Since then, work in this field has continued unabated.

Studies on natural language processing and especially information extraction in Turkish; Compared to studies on texts in other languages such as English, German and French, it is quite limited in number and scope. However, a considerable amount of documents are created in Turkish every day, and the automatic extraction of the information in these documents is very important for systems such as automatic information access, question answering and summarization. Turkish language has a complex structure in terms of its morphological features. For this reason, there are a limited number of studies on Entity Name Recognition for Turkish.

For years, different methods have been studied in entity name recognition. Some of them are as follows: Rule Based Systems, Conditional Random Fields, Decision Trees, Hidden Markov Model, LSTM and BERT.

Recently, Name Entity Recognition models have started to predominantly use machine learning approaches. Küçük and Yazıcı (2010, 2012) transformed their rule-based system into a hybrid system for better performance when applied to different domains. Yeniterzi (2011) developed a NER system based on Conditional Random Fields with a dataset created by using morphological analysis. Other Conditional Random Fields based systems were proposed by Özkaya and Diri (2011) and Şeker and Eryiğit (2012). It was seen that Şeker and Eryiğit (2012) systems achieved better results when compared to other projects using the same data set. Demir and Özgür (2014) developed a system with a neural network-based semi-supervised approach. Although this system was created without an annotation file in the dataset, it achieved high results from the Şeker and Eryigit(2012) system.

The BERT model was created and published by Google in 2018. BERT had high success rates in the languages in which it has been using since then. This model aims to extract meaningful information by evaluating the sentence in two directions (from right to left - from left to right). Since the studies for Turkish are limited, it aimed to conduct

a study for this new technology on the Turkish language. In the form of the dataset, we used words and entity tags. The goal of this study is to build a system for extracting the information from a text document for the name entities such as organization, person, time, place, date, money, percentage.

II. BERT Model

In the model that we created, the Named entity recognition task for Turkish was created using the BERT model. In the BERT model that we use for NER, we first put the model in pre-training so that it can understand the structure and semantic integrity of the Turkish language. In the pre-training part, two methods called Masked Language Modeling and Next Sentence Prediction are used.

There are two-phase in the BERT model:

- Pre-training.
- Fine-tuning.

BERT model completes the pre-training by applying two methods. Those methods are Masked Language Modeling and Next Sentence Prediction methods.

Masked Language Modeling

Masked Language Modeling aims the model to fill these gaps in the sentence by masking some words in the incoming sentence. For instance

In the next sentence there are some words that is masked:

The [MASK1] brown fox [MASK2] over the lazy dog. [MASK1] = quick [MASK2] = jumped

This method helps the model to learn the bidirectional reading structure of the sentences.

Next Sentence Prediction

In the next sentence prediction method, to predict the next word, BERT takes two sentences and decides whether the second of these sentences is a sentence that follows the first. Model can learn the semantic meaning of the language with next sentence prediction method. With the next sentence prediction our model can understand the relation between sentence pairs. For instance,

Sentence A: Turn the radio on.

Sentence B: She bought a new hat

In the preceding sentence pair, sentence B is not a follow-up of sentence A. So, those sentences will be labeled as notNext.

With the help of these the methods, BERT model can be fine-tuned for specific tasks with the help of language that is learned in the pre-train phase.

In the BERT model we need input into embeddings to do that, model uses three embedding layers indicated here:

- Token embedding.
- Segment embedding.
- Position embedding.

Token embedding is pre-trained embedding model and the base model includes a word vectoring piece with 30k tokens. Segment embeddings is the vector that holds the sentence number of the embedded sentence. Position embeddings holds position of a word in an embedded sentence.

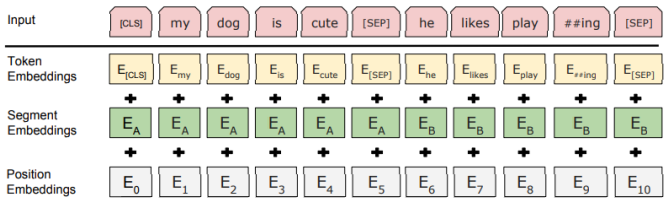

Figure 1 Embedding Layers[1]

Fine Tuning for BERT

In the case of entity name recognition, we need to train the fully connected output layers by transforming them into output layers that will predict the class for the completely new given word, and then train these layers with a dataset created for the entity name recognition model using supervised learning.

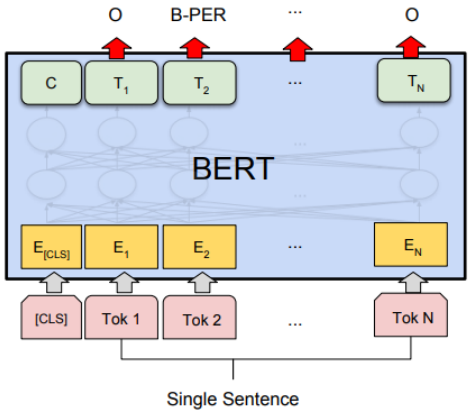

Figure 2 Fine Tuning for NER[1]

The training time is fast as the new learning parameters of the model are only the output layers created in fine tuning . For layers in the pre-train phase, there are no major changes in the parameters. Supervised learning techniques are used to get named entities for our input with the help of training the model with the dataset created for named entity recognition. The model that is pre-trained for Turkish with the addition of output layers for the NER task can obtain a Turkish NER model.

III. NER System

This project has been developed in python language because of the flexibility of its Natural Language Processing libraries and its strong infrastructure for working on data.

There are two columns in the dataset used to train the system: entity and name. This data is transformed into a dataframe. Every tuple of the dataframe includes a sentence id, name, and named entity tag.

The frontend was built using Flask, a Python framework. The data was read into the dataframe and processed quickly thanks to the flexible structure of the language. In the dataset, only the word and its entity tag are maintained. The dataset's missing data was removed. The parameters for training the system is mentioned below:

$$
\begin{aligned}
&\text{args} = \text{NERArgs}() \\
&\text{args.num}_{t}rain_{e}pochs = 12 \\
&args.learning_{r}ate = 1e-4 \\
&args.overwrite_{o}utput_{d}ir = True \\
&args.train_{b}atch_{s}ize = 8 \\
&args.eval_{b}atch_{s}ize = 8
\end{aligned}
$$

The model was given predetermined entity tags and parameters, which were used in the training phase. This parameters used for initializing the BERT model. The BERTurk model, which is available in Turkish, was used in the project's development. With a 35GB training corpus, this model was trained using the Google TensorFlow Research Cloud system.

BERT model uses the encoding and decoding structure. When a sentence came out the system our model encoder learns how to represent the data which is mentioned above.

For fine-tuning the berTurk model which is a pre-trained model for Turkish Language, when an input is given to the system, input sentence will be divided and every word is called tokens. For each token, Bert model will add a [CLS] and [SEP] token. System will feed those token representations to a classifier (feedforward network + softmax function). Then, the classifier returns the category to which the named entity belongs. [2]

The user will see the input text as well as the named entities as an output once the model has labeled the names. If a user wants to contribute to active learning, they can provide a dataset in the format described in the following section. The system will add the users' dataset to available dataset. After this process is done, system will train itself for the new dataset. Another way to do active learning for the system is if a user wants to add a system to the data set, the system will do it automatically.

IV. Data Set

There are several ways to represent dataset. The way we choose to represent our data has a huge impact on the performance of NER. Simplest method to represent dataset is raw tag method but with this method there is no way to show consecutive connected labels in dataset. The most common representation scheme for tagged entities is the IOB2 representation(Tjong Kim Sang 2002) (also known as BIO). The IOB2 scheme is a variation of the IOB. With this representation, the first token is prefixed with "B-" on any named entity tag type. For multi-token labels, the following labels use the "I-" prefix. Tokens are labeled with an 'O' tag if they do not belong to one of the predefined entities. The training and test data used in the training phase of our model merged from various news reports and academic studies. Merged data tagged with IOB2 representation schemes. In our dataset we used 7 named entity classes to train our data. Those tags are:

- Organization, Location, Date, Money, Percent, Time and Person

Number of samples in each class type can be found in followed table.

Class	ORGANIZATION	LOCATION	DATE	MONEY	PERCENT	TIME	PERSON
Dataset (13.9 MB)	63240	46180	4871	3613	2610	589	72939

Figure 3 Number of samples for each class

Our dataset is labeled with IOB2 tagging scheme within this format showed in the table.

Token	Raw tag	IOB2 tag
Mustafa	PERSON	B-PERSON
Kemal	PERSON	I-PERSON
Atatürk	PERSON	I-PERSON
23	DATE	B-DATE
Nisan	DATE	I-DATE
1920	DATE	I-DATE

Figure 4 Raw and IOB2 tagging [3]

"Mustafa Kemal Ataturk" is the full name of the father of Turks. It starts with the "Mustafa" so it is labeled with "B-" infix consecutive names that follow "Mustafa" are labeled with "I-" infix since those names are parts of the whole name.

V. Performance Analysis

Hyperparameters such as the number of epochs can affect a model in a good way or in some cases it affects in a bad way. To understand the optimum epoch number for our model, we make some performance analyses. The fact that the small number of epochs can not show the model's real f-score however after some boundary value the model starts to understand data too well. This happens when we train our model too much and that can have some negative effects. To understand when our model will memorize the dataset, we can check the results for models that are trained with the different number of epochs. The result of those training can tell us where our model will reach it is maximum performance.

According to these results we obtained, although the results obtained from 1 and 3 epoch numbers are great outcomes. It was observed that the precision, recall, and f score for the model is increased when the model is trained with an increased number of epochs. However when the model reaches a point where epoch number can not improve the performance of the model. This can be seen from the table that from 8 epoch to 12 epochs, the model gain 0.007 precision but also it lost 0.001 recall and the f-score

	1 Epoch	3 Epoch	5 Epoch	8 Epoch	12 Epoch
Precision	0.887	0.90	0.905	0.906	0.913
Recall	0.845	0.85	0.859	0.863	0.862
F-score	0.866	0.878	0.881	0.884	0.887

Figure 5 Results for model trained with different epoch numbers.

increased just by 0.003. According to these results, it can be said that after 12 epoch model will reach a point where it can not improve anymore. After 12 epochs there is a great chance that the model will memorize the dataset and it will reach a point that it can not predict any words other than its train dataset.

VI. Experimental Results

In this section, the Results of the BERT model that is trained using our main dataset will be compared for some entities. Extreme cases like "TIME" entity labeling were tested. Labeling words with the "TIME" entity is an extreme case since our dataset has a small number of "TIME" labels. In those conditions, it can be concluded that labeling words with "TIME" is an extreme case.

In the training section, Training is done with two separate datasets one of them is the main dataset that is used in the main model other one is a dataset that is created at the start of our project.

	Result of a model that is trained with big dataset	Result of a model that is trained with small dataset
Precision	0.91	0.88
Recall	0.86	0.83
F-score	0.89	0.85

Figure 6 Result for models with different training data size.

With this results it can be shown that training a model with different sized dataset can make major differences in recall, precision and f-scores.

Model's prediction result for the input sentence:

11,B-DATE), (Haziran,I-DATE), (günü,O), (saat,O), (8,B-TIME), (:,I-TIME), (30,I-TIME), (civarlarında,O), (Göktug,B-PERSON), (ve,O). (Aslı,B-PERSON), (ara,O), (proje,O), (konularına,O), (çalısmaya,O), (basladılar,O), (12,I-TIME), (:,O). (00,I-TIME), (saatinde,O), (çalısmayı,O), (bitirdiler,O), (Bir,O), (sonraki,O), (toplantının,B-DATE), (yüzde,B-PERCENT), (50,I-PERCENT), (lik,O), (kısmınının,O), (planı,O), (da,O), (yapıldı,O).

This sentence was chosen for the prediction job because it contains several words that are labeled with extreme entities, such as "TIME" and "PERCENT," as well as certain words for which the model has been trained using a large amount of data.

VII. Conclusion

In this paper, we have briefly mentioned that the usage of the BERT model to create a Named Entity Recognition system for the Turkish language that provides better output to contribute the literature, and when the results of the BERT model compared with its pairs, it is clear that BERT can have great results for Named Entity Recognition task.

The project has been carried out in such a way that it is ideal for developing the dataset for users who wish to contribute and develop the data set based on user input. As a result, the system is designed to grow on its own.

It is expected that the project will contribute to natural language processing studies in Turkish because it is intended as a system open to development and growth during the development phase. The project might benefit from user correction of wrongly labeled words. Simultaneously, increasing the size of the dataset will improve accuracy.

As future work, we are going to create an interface for users that they will type their sentences into the system also they can change the label of some words so that our dataset can grow more accurately.

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[2] S. Ravichandiran, *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT*. [Online]. Available: https://books.google.com.tr/books?id=CvsWEAAAQBAJyear=2021, publisher=Packt Publishing

[3] K. Oflazer and M. Saraçlar, *Turkish Natural Language Processing*, ser. Theory and Applications of Natural Language Processing. Springer International Publishing, 2018. [Online]. Available: https://books.google.com.tr/books?id=D-5lDwAAQBAJ