BACKGROUND LINKING ON NEWS ARTICLES

A PREPRINT

Çağhan Köksal
Department of Computer Science
Sabanci University
Tuzla/İstanbul, 34956
caghankoksal@sabanciuniv.edu

Ali Eren Ak
Department of Computer Science
Sabanci University
Tuzla/İstanbul, 34956
akali@sabanciuniv.edu

Buse Çarık
Department of Computer Science
Sabanci University
Tuzla/İstanbul, 34956
busecarik@sabanciuniv.edu

Ercüment Yıldırım
Department of Computer Engineering
Middle East Technical University
Çankaya/Ankara, 06800
ercument.yildirim@metu.edu.tr

Reyyan Yeniterzi
Department of Computer Science
Sabancı University
Tuzla, Istanbul 34956
reyyan@sabanciuniv.edu

August 26, 2020

ABSTRACT

In recent years, people have accessed and shared information through online web services. Nevertheless, most of the authors of the articles assume that all readers have the same background knowledge. For several readers, this shortage of information leads them not to understand the entire context. This issue leads to help readers to understand the text by retrieving articles that provide background and contextual information to the readers about the news they read. In this report, we address this problem in the context of background linking and wikification tasks of the TREC 2020 news track. This issue was also discussed for the Turkish language in terms of the background linking task. As a first step, an attempt was made to identify named entities carrying the contextual information of the articles by using CRF and BERTurk. The results on both Turkish NER and SuDer corpora illustrate that transformer-based models surpass statistical approaches.

Keywords Background Linking · TREC · Named Entity Recognition · Information Retrieval

1 Introduction

Online news services have been the primary source for sharing and receiving information among people for many years. Although the common assumption is that the readers have the necessary background knowledge about the context and entities mentioned in the article, this shared expectation may not be valid for many readers. Hence, the need for providing information has driven the urge to help readers understand the news they are reading in terms of background and context. However, the detection of articles that contain background context is not a trivial task. Therefore Text Retrieval Conference (TREC) introduced two tasks to provide solutions for this problem; these tasks are background linking and wikification. The purpose of the background linking task is to retrieve the articles that help the readers to understand the context and obtain background knowledge of a given article (the query article). The wikification task addresses this problem by connecting to Wikipedia links that yield more information on entities and concepts in a given text. This contextual knowledge of the articles is carried over by named entities, which are real-world objects such as person, location, or organization names. Hence, to achieve the tasks mentioned above, the acquisition of these named entities has vital importance. Named entity recognition (NER) identifies and classifies these named entities by assigning predefined category tags such as a person or organization name. Different approaches have been studied for the NER task, where the statistical and neural architectures are the popular methods. CRF is a widely used statistical model for NER, which models the dependency between features. Although the results of CRF models today show remarkable success, neural network structures outperform other approaches. In neural architectures, commonly used BiLSTM models have been replaced with transformer-based models for sequence-to-sequence problems in recent years since transformers capture the contextual information and long-range dependencies (Vaswani et al., 2017). Bidirectional

Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018) make predictions only from the context of the words through masked language models that find the label of randomly masked words in a sentence. Also, it can be used for a specific task by fine-tuning of a pretrained model.

In this study, firstly, submissions for two TREC tasks were achieved. The aim of the other side of the project is to apply the background linking task for the Turkish news articles. For the Turkish language, primarily, a new dataset was started to be collected with RSS feeds of different news sources. Afterward, since NER is the first step of the background linking problem, this task has been studied by training with CRF and transformer-based models on the Turkish NER dataset. The current state-of-the-art result for the Turkish NER task was reproduced with Bidirectional Encoder Representations from Transformers with a CRF at the top layer. As the final phase, the SuDer corpus was used for the background linking task.

The rest of the report is organized as follows.Section 2 summarizes the related studies on background linking and named entity recognition. Section 3 describes our approach in both Turkish and English NER and background linking tasks. Section 4 details the dataset used in this study. Section 5 discusses our experimental setups and results, and Section 6 concludes the report and presents our future work.

2 Related Works

2.1 TREC

The Text REtrieval Conference (TREC) is a program sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense. It was started by 1992 and with the aim of encouraging the information retrieval (IR) community to research in that field. Every year researchers are invited to participate in many tasks[1] such as Conversational Assistance Track, Decision Track Results, Deep Learning Track, Precision Medicine Track etc. In last year's News Track in Text Retrieval Conferences (TREC) is continued with two significant tasks about information retrieval problems in news articles: background linking and entity ranking tasks. Last year, 28 runs from 9 participation and 22 runs from 8 participation were submitted for Background Linking and Entity Ranking tasks respectively (Soboroff, Huang, & Harman, 2019).
In the Background linking task, the approach focusing on estimating the entity weights according to their context gets the highest result in the median of nDCG@5 score (Lu & Fang, 2019). This work indicated that entities that have a Wikipedia page tend to have more contextual information about the article when it is used in background news article retrieval. Besides that, with their two submissions, it is also examined to the importance of the entity according to their context. This means the entity which is widely used throughout the article, is more significant than the entities which are generally used in a subpart of text. For future work, they indicated it could be useful to analyze and improve the effect of contextual entities and also regular time filters.
On the other hand,the approach based on examining the effect of the unsupervised Doc2Vec model (Le & Mikolov, 2014) on entity ranking tasks leads to the last year's task. The cosine similarity between the entities' Wikipedia page and article's vectors is the baseline for this approach. Therefore in addition to this baseline entities are ranked using other background linking articles. Also, Doc2Vec is used to acquire background linking articles, however; it could be argued that Doc2Vec may not be useful methodologies for background linking when considering their results did not overperform baseline (Fayoumi & Yeniterzi, 2019).

2.2 Doc2Vec

Paragraph Vector is an unsupervised algorithm that takes variable-length texts and learns fixed-length feature representations from them. It was introduced in 2014, by researchers from Google as a strong alternative to bag-of-words and bag-of-ngrams models which are vector representations used in text classification and clustering. There are two major strengths of the Paragraph Vectors. First, they keep the semantic of the words. And second, they consider the order of words. The state-of-the-art results were obtained on many tasks by using the Paragraph Vectors (Le & Mikolov, 2014).

2.3 NER in Turkish Language

Named entity recognition, has studied in Turkish for many years (Küçük, Arıcı, & Küçük, 2017). HMM model is the first statistical approach for Turkish NER, which deals with the agglutinative form of the Turkish language by utilizing lexical and morphological information of words.(Tür, 2000) The first rule-based approach was introduced with information resources such as the dictionary of person names or well-known entities, and pattern bases. (Küçük et al., 2009). In another work, the first CRF model was presented for the Turkish NER task. Also, this study used morphological features and roots as separate tokens instead of words to emphasize the impact of morphological and syntactic characteristics of the Turkish language on this task.(Yeniterzi, 2011) An automatic rule-based learning

[1] Last year TREC tasks and results: `https://trec.nist.gov/pubs/trec28/trec2019.html`

procedure was applied and utilized morphological features(Tatar & Cicekli, 2011). (Şeker & Eryiğit, 2012) examined the effects of using morphological features of Turkish language as a feature of Conditional Random Field on Named Entity task. Additionally, they demonstrated contribution of using gazetteers on the task.Moreover, they compared the results of previous works on CoNLL and MUC evaluation metrics which is important contribution; since in recent works different evaluation sets and metrics were used and average score of them is reported.
Besides statistical and rule-based approaches, neural structures have gained popularity in recent years. The first study in the neural-based model employed neural networks in the semi-supervised method with language-independent features such as previous tags and word capitalization patterns and retrieved continuous word representations(Demir & Özgür, 2014) Instead of word embeddings, using only the sequence of characters to represent the words in a stacked bidirectional LSTM model reached a result close to the other BiLSTM models.(Kuru, Can, & Yuret, 2016) One of the commonly used structures in sequence-to-sequence tasks is bidirectional long short-term memory (BiLSTM) is applied for the Turkish NER. In recent work (Güngör, Güngör, & Üsküdarli, 2019), the BiLSTM model is presented accompanied by CRF as the top layer and the concatenation of word, morphological, character embeddings as the input. Another well-performed architecture, Transformer based models, firstly studied in both cased and uncased models (Schweter, 2020). The current state-of-the-art performance in the Turkish NER task is obtained with BERTurk completed with CRF as the top layer.(Aras, Makaroglu, Demir, & Cakir, 2020) This study also compared the effect of different initialization for word embeddings.

3 Methods

3.1 Background Linking

For Background Linking task, we used 3 different approaches. In addition to full-text query which is our baseline approach, we have examined the use of transfer learning by using Universal Sentence Encoder model and BERT Text Summarization.

3.1.1 Baseline - Bag of Words

As a baseline approach, we used full-text as our search query. Elastic Search applies same processing steps that are used in indexing step. Moreover, we have applied regular date filter which brings documents that are published before the query document.

```
 {
    "bool":{
        "must": [
            {
            "range":{
                    "date": {
                    "lt": query["date"] #less than    }
                     }
            },
            {
            "multi_match":{
                    "query":query["raw_text"],
                    "fields":[
                    "text",
                    "author",
                    "title",
                    ]
        }}]}
}
```

Listing 1: Full Text Query Example

In the listing 1, by using must with bool tag, we retrieved the documents whose date field value is smaller than the query document. "lt" tag stands for less than. Raw text which is whole news document, is given as a query and by using multi_match tag, we specified multiple fields to search our query.In this example, "text","author","title" are the fields that we searched.

3.1.2 Universal Sentence Encoder

(Cer et al., 2018) presented 2 models(Deep Averaging Network and Transformer based) that encodes sentences into 512 dimensional dense vector embeddings. We used the Universal Sentence Encoder 2[2] and converted our Washington Post(WaPo) documents into embeddings. We have indexed the dense vectors as a field on ElasticSearch. While we were querying the documents, we compared the cosine distances between them and retrieved the documents which have smaller cosine distances. With this approach we examined the use of transfer Learning and effectiveness of embedding approaches in Background Linking Task.

3.1.3 BERT Summarization

Automatic text summarization in NLP is a significant problem with the challenge of creating a fluent summary while keeping crucial information in article meaning. Researchers examined the problem from various approaches in recent years. However, all approaches are utilized mainly in two bases in this task: Abstractive and Extractive summarization. Abstractive Summarization is generating new, short, and paraphrased versions of the original text. While human summarization is mostly similar to abstractive, extractive summarization is generally used in NLP. Because extractive summarization generally overperforms in many tasks and significant problems in abstractive summarization exist such as semantic representation and natural language generation (Allahyari et al., 2017).

In this approach, it is aimed to eliminate minor arguments and to focus on the main arguments with extractive summarization. Bert Extractive Summarization that was fine-tuned by Miller (Miller, 2019). In this model, sentence embeddings run with a clustering algorithm which is used to get more close sentences to clusters centroids. In addition to sentence embedding, coreference resolution techniques are also used to get more contextual words in the text. Contextually more important sentences are extracted and used for regular queries in elastic search. In each news article in 2020 topics given by TREC have 860-words averagely. However, it is decreased to averagely 180 words when it is summarized by the BERT model. Consequently, It is indicated that other words that are not considered in summarization do not have contextual information about the article as much as sentences in a summarized text.

3.2 Wikification

This year's tasks changed to entity linking from entity ranking differently from last year's. This means that participants should also extract entities from articles in addition to ranking them. Therefore, in our both two approaches, the Stanford NLP [3] framework was used to recognize entities in articles. However, in addition to frameworks NER pipeline, statistical techniques are also used to filter and rank entities.

Before linking entities to corresponding wiki-page, they are extracted with Stanford Corenlp NER pipeline. After that firstly, to filter entities, a simple entity-tag filter is applied. It is explored that nearly half of the extracted entities have numerical and temporal expressions while generally tend to be less important in the given article. Secondly, it is examined that pronouns are also considered as an entity by Stanford Core NLP, therefore they are also filtered to extract more relevant entities about context. Thirdly, duplicated entities, even though they might be linked by TREC separately, are filtered by their position in the article. For example, "Donald Trump" can be referenced in a news article with different types such as "President Trump", "Trump" or "Candidate Trump", etc. However, except "Donald Trump", others are filtered according to their word count, phrase match, and position. Lastly, the entities started with a lowercase letter are simply filtered because proper nouns, which generally starts with upper case letters, are trying to be extracted in this case.

Moreover in this task, entities should be linked to the corresponding Wikipedia page. In this step, our two approaches are differentiated. Firstly entities simply ranked according to their position and linked to the corresponding wiki-page with Elastic Search. Wikipedia pages are indexed with the page title, article, categories, and names; however, title match with entities is considered in the Elastic Search queries. Secondly, last year's Doc2Vec model is also used in addition to baseline. Due to change in the task, it could be hard to compare the model with last year's topics; however, in addition to our background linking submission, the capability of Elastic Search in the wikification task is examined.

As Fayoumi (Fayoumi & Yeniterzi, 2019) indicated, Doc2Vec may not be an efficient approach in background linking while it has led last year's entity ranking task (2019). Therefore the combination of Doc2Vec and Elastic Search could be examined in both approaches. In addition to that Stanford Core NLP has an entity linking pipeline which could be examined for future works. Due to the time limit, it could not be examined in this study.

3.3 Turkish RSS Feed Data Collection

As the first step of this study, a new dataset consisting of Turkish news articles has begun to be collected from online news websites. These sources were clustered into two groups according to whether the news articles were divided

[2] https://tfhub.dev/google/universal-sentence-encoder/2

[3] https://stanfordnlp.github.io/CoreNLP/

into categories or not. Hürriyet [4], Ahaber [5], BBC Türkçe [6], Mynet [7], NTV [8], Sabah [9], Star [10], Takvim [11], Vatan [12], and Yenişafak [13] are the sources that are grouped under sections. These resources are gathered in 10 different groups on average such as economy, world, health, education, sport, and technology. There are only three news sources without categories, namely Sözcü, Cumhuriyet, and Donanım. The information in the news was obtained from the RSS feeds on the websites of sources. From these feeds, title, date, category if they have, and link of the articles were driven by the Feedparser library in Python. In order to extract the full-text article by an RSS link, a third-party API service of a website [14] is used through requests library. When full-text of news is acquired, the collected text is preprocessed to eliminate erroneous articles and advertisements by the determination of patterns of these misleading contexts in each source. After retrieving all information from utter resources, each news content is written to the file in JSON format as the source, category, title, date, and text of the news as in the example 2. End of 5 hours following the printing process of the collected data, the algorithm restarts to the entire process to obtain new news articles.

```
{
    "source": "ntv",
    "category": "egitim",
    "date": "Mon, 27 Jul 2020 04:04:09 +03:00",
    "title": "Bakan Selcuk'tan okullar acilacak mesaji",
    "link": "https://www.ntv.com.tr/egitim/bakan-selcuktan-okullar-
        acilacak-mesaji,qAi5oIHoJEWwiiGtKsLUqg",
    "content": "Milli Egitim Bakani Ziya Selcuk, okullarin gerekli
        hazirliklar yapilarak acilacagini ..."
    }
```

Listing 2: RSS Feed Data Example

3.4 Turkish Morphology

Turkish is an agglutinative language with word structures formed by adding morphemes to root words. There are many irregularities in Turkish caused by some phonetic rules. Since there can be a lot of possible meanings for one word in Turkish, a detailed morphological analysis should be done to get necessary information for syntactic analysis and semantic analysis(Oflazer, 1994).

3.5 Named Entity Recognition

3.5.1 Zemberek

Zemberek(Akın & Akın, 2007) is an open source NLP Framework that provides tools and models for morphological analysis, morphological disambiguation, named entity recognition, tokenization etc. We used Zemberek in order to tokenize our corpus.In addition to that, we have used Zemberek's morphological analysis tool to get morphological analysis of the tokens.We utilized the morphological analysis and extracted features to use as a feature for Conditional Random Field.

[4] https://www.hurriyet.com.tr/
[5] https://www.ahaber.com.tr/
[6] https://www.bbc.com/turkce
[7] https://www.mynet.com/
[8] https://www.ntv.com.tr/
[9] https://www.sabah.com.tr/
[10] https://www.star.com.tr/
[11] https://www.takvim.com.tr/
[12] http://www.gazetevatan.com/
[13] https://www.yenisafak.com/
[14] http://ftr.fivefilters.org/

Sent ID	token	Morp. Analysis
5506	Anadolu	anadolu+Noun+Prop+A3sg
5506	Medeniyetleri	medeniyet+Noun+A3pl+P3sg
5506	Müzesi	müze+Noun+A3sg+P3sg
5506	’	+Unk
5506	nin	nin+Unk
5506	Çatalhöyük	çatalhöyük+Noun+Prop+A3sg
5506	bölümünü	bölüm+Noun+A3sg+P3sg+Acc
5506	gezerken	gez+Verb+Aor^DB+Adv+While
5506	elde	el+Noun+A3sg+Loc
5506	edilen	ed+Verb^DB+Verb+Pass^DB+Adj+PresPart
5506	...	...

Table 1: Morphological Analysis of tokens

In the table 1, each token we have its corresponding morphological analysis.

3.5.2 NER Data Preprocessing

In the NER Dataset, named entities start with <b_enamex TYPE="ENTITY TYPE"> tag and end with <e_enamex> tag. By using regex, we have find the named entities, converted them into IOB tagged format. First token of the named entity starts with "B-" tag and corresponding entity types comes after it. If the named entity consists of 2 or more words, first token starts with "B-" tag and other tokens take "I-" tags thats stands for INSIDE. With this step, we converted our NER dataset which is in ENAMEX format into IOB tagged sequence format to apply various sequence models.

```
{
<b_enamex TYPE="ORGANIZATION">Anadolu Medeniyetleri Muzesi<e_enamex> ’nin
    <b_enamex TYPE="LOCATION">Catalhoyuk<e_enamex>  bolumunu gezerken elde
    edilen bulgularin cok onemli kesiflere dayandigini da goruyoruz.
}
```

Listing 3: ENAMEX format

In the listing 3, example of ENAMEX format can be observed.For instance,Diyarbakır which is named entity with LOCATION type, is surrounded by <b_enamex TYPE="LOCATION"> and <e_enamex> tags.

Sent ID	token	IOB tag
5506	Anadolu	B-ORGANIZATION
5506	Medeniyetleri	I-ORGANIZATION
5506	Müzesi	I-ORGANIZATION
5506	’	O
5506	nin	O
5506	Çatalhöyük	B-LOCATION
5506	bölümünü	O
5506	gezerken	O
5506	elde	O
5506	edilen	O
5506	...	...

Table 2: IOB Tagged Data Format

In the table 6, tokens which are converted into IOB tagged sequences can be seen.

3.5.3 Conditional Random Fields

Conditional Random Fields(CRF) (Lafferty, McCallum, & Pereira, 2001) are discrimitative probabilistic model that is used for labelling and segmenting sequence data. CRF has several advantages over Hidden Markov Models(HMM) and maximum entropy Markov Models(MEMMs) such as modeling dependencies between features and solving label bias problem. Since CRF let us use various features both from previous and future time-step, we have examined effect of several features on NER task.Sklearn-crfsuite[15] framework which is open source CRF sequence labelling toolkit is used

[15] https://sklearn-crfsuite.readthedocs.io/en/latest/

in CRF training and for evaluation we have used the CoNLL evaluation script[16] and seqeval framework[17] is used. We have used Python programming language.

Features We have used morphological and lexical features of the words in the sentences. Eventough, CRF gives freedom for using many features, we limited the number of used features by only creating features for previous token, current token and the next token.

word.lower() : We have used lowercase version of word.Since Turkish is agglutinative language, using stem version of the word causes sparse data problem. Therefore, we have used surface form of the word.

word.postag(): Part of speech tag(POS) of the word, after the last derivational boundry is given as a feature to CRF. Since Turkish is agglutinative language, suffixes that added to the words might change the meaning and Part of Speech(POS) tag of the word completely, therefore; we have used POS tag after the last derivational boundary.

word.nounCase(): This is binary feature for the case argument. It is 1 if morphological features of the word contains Nomitative(Nom), Accusative(Acc), Dative(Dat) , Ablative(Abl), Locative(Loc), Genitive(Gen), Instrumental(Ins), Equative(Equ) tags. If not it is 0.

word.INF(): All inflectional tags after the derivational boundry given as string.

word.istitle(): A binary feature that indicates whether all the words in the sentences startswith an uppercase letter or not.

word.isupper(): A binary feature that indicates whether all characters in the word are uppercase or not.

word.isdigit(): A binary feature that indicates whether all the characters in the word are digits or not.

BOS: A binary feature that indicates whether the word is in the beginning of the sentence or not.

postag[:2]: First 2 characters of the POS tag.For instance if POS tag of the word is Noun, No is given as feature to CRF.

+1:word.lower(): Surface form of the next token.

+1:word.istitle(): A binary feature that indicates whether all the tokens in the string starts with an uppercase letter. +1 implies that this feature belongs to the next token.

+1:word.isupper(): A binary feature that indicates whether all characters in the word are uppercase or not. +1 implies that this feature belongs to the next token.

+1:word.postag(): POS tag of the next token.

-1:word.lower(): Surface form of the previous token.If word is the start of sentence it is None

-1:word.istitle(): A binary feature that indicates whether all the tokens in the string starts with an uppercase letter. +1 implies that this feature belongs to the previous token. If word is the start of sentence it is None

-1:word.isupper(): A binary feature that indicates whether all characters in the word are uppercase or not. -1 implies that this feature belongs to the previous token.If word is the start of sentence it is None

-1:postag: POS tag of the previous word. If word is the start of sentence it is None.

-1:postag: First 2 characters of the POS tag of the previous token.

word[-3:]: Last 3 characters of the current token.

word[-2:]: Last 3 characters of the current token.

3.5.4 BERTurk NER

Transformers have replaced convolutional and repetitive neural network models in natural language processing problems in recent years since they seize the long-range sequence features (Vaswani et al., 2017). Model pretraining, which is also contributed by Transformers, facilitates models to train on a large corpus and easily used for specific tasks (Wolf et al., 2019). One of the transformer-based models, Bidirectional Encoder Representations from Transformers (BERT)(Devlin et al., 2018), applies a masked language model pretraining, which allows it to learn from only its context contrary to the unidirectional language model pretraining. The masked language models masked words randomly from

[16] https://github.com/sighsmile/conlleval/

[17] https://pypi.org/project/seqeval/

a given sentence instead of predicting the next token as in the other language models(Devlin et al., 2018). In this study, Turkish BERT (BERTurk) [18] was used as the pretrained masked language model and fine-tuned it for the NER task on the Turkish NER dataset. Firstly, the data were transformed into CoNLL format, in which each line consists of a token and its corresponding label, and different sentences are separated with an empty line. As our data is already separated by tokens, the BERT tokenizer only operates WordPiece tokenization, which uses subwords instead of words to utilize the common sub-parts (Devlin et al., 2018). This approach also alleviates the data sparsity problem, which is common in morphologically rich languages such as Turkish (Aras et al., 2020). Implementations were performed in the HuggingFace transformers library [19] (Wolf et al., 2019).

4 Datasets

4.1 NER Dataset

Turkish NER Dataset (Tur, Hakkani-Tur, & Oflazer, 2003) contains 28K sentences and 500K words, including 24K person names, 16K location names and 14K organization names which are labeled with enamex format.

4.2 TREC Dataset

```
{
    "id": "9171debc316e5e2782e0d2404ca7d09d",
    "article_url": (...),
    "title": (...),
    "published_date": 1472713234000,
    "contents": [content:{...}, content:{...}, ... , content:{...}],
    "type": "blog",
    "source" : "Washington Post"
    }
```

Listing 4: Json Line Example

TREC's this year's News Track provides Washington Post Collection with 595,037 news articles from 2012 to 2017 for participants in tasks. Data is the same as last year's data in addition to the cleaning process of duplicate documents. Each news article is stored in the "JSON-lines" format which has structured data as could be seen in the figure 4. "Contents" block include "content" items for each sentence in the article. Content block also includes the "type" field to filter if the content is not really an article. More detaily, if the type is "sanitized_html", content includes article sentences. Otherwise, it does not. Also, the contents block includes "url" item to indicate the corresponding page of news articles and multimedia.
Besides Washington Post Collection, CAR-track formatted Wikipedia articles by August 2017 are also provided by TREC. Although it is mainly provided for entity ranking tasks, many submissions in last year's background linking use Wikipedia data in their approaches. Also, TREC lets participants free to use Wikipedia data in the background linking task. However, it was preferred to use only the Entity linking task in our submissions.

4.3 SuDER Dataset

(Sen & Yanıkoglu, n.d.) collected approximately 426K[20] news pages from Sabah website [21] between January,2010 and July,2017. They eliminated the pages that have less than 10 words. After this process, they have 420513 documents with 4 categories which are "gündem"(agenda), "yaşam"(life) , "ekonomi"(economy) and "yazarlar"(authors) In addition to that, they collected approximately 463K articles which had released before September,2017.Unlike Sabah, only 273K document have category information. In addition to elimination of documents that have less than 10 words; categories that do not have enough documents are eliminated.After these processes, there are 268784 remaining documents and 14 different categories.(Sen & Yanıkoglu, n.d.) While (Sen & Yanıkoglu, n.d.) used the dataset for classification task, we will use the dataset for information retrieval tasks.

4.4 SuDER NER Test Dataset

In order to interpret correctness of trained NER models, we have created another test dataset by labeling 40 news from SuDER corpus. Randomly 40 news are selected by applying minimum word count threshold. The test data consist of 815 Location 644 Organization and 811 Person entities.

[18] https://huggingface.co/dbmdz/bert-base-turkish-cased/

[19] https://github.com/huggingface/transformers

[20] K stands for 1000

[21] https://www.sabah.com.tr/

5 Results

5.1 TREC Results in NDCG@5

	2018	2019
Baseline	0.5623	0.4049
USE	0.4145	0.3308
BERT-Summarization	0.5288	0.3351

Table 3: NDCG@5 Results of Background Linking

TREC's this year's News track has not resulted yet; however, according to the previous year's results could give presumption about the performance of approaches. When considered that other approaches did not overperform baseline in both 2018 and 2019 years results, it could be argued that Universal Sentence Encoder (USE) Embeddings and BERT Extractive Summarization may not be proper models for background linking. However, they are experimentally valuable due to their contributions. For instance, BERT Summarization score is close to baseline when it is considered as searching key sentence query in elastic search, and also it is one of the first BERT approaches in background linking. Using BERT on the Background Linking task will give insight about the performance of the state-of-the-art language models and use of transfer learning for future works. On the other hand, it could be argued that use of document embeddings in background linking is also valuable for future work. Last year, the highest scores on the entity ranking task was achieved by using Doc2Vec which shows that more advanced document embedding techniques may improve the score.

5.2 NER in Turkish

5.2.1 BERT vs CRF ve Zemberek

	Turkish Ner Dataset	SuDER Test
BERTurk	0.94	0.88
CRF	0.92	0.74
Zemberek	0.90	0.75

Table 4: BERT vs CRF F1 measure comparison in CoNLL

Firstly, our three models were evaluated with the Turkish NER dataset, which was split as train, validation, and test sets corresponding ratios %80, %10, and In order to compare the generalization of our models, both Zemberek, CRF, and BERTurk models were evaluated on an external dataset that is labeled 40 news from the SuDer corpus. The results are given in the right column of Table 4. From the previous results, it was expected that BERTurk demonstrates a better performance. However, the BERTurk improved the results of other models significantly by roughly 14 percent. The reason for this difference is that a considerable amount of words from the Suder corpus are not contained in the training dataset since the Turkish NER news articles include older news than the SuDer corpus. These lead to different agendas and different person and organization names in these datasets. This separation in corpora causes a drop in joint named entities. The reason CRF cannot map to dependencies between features well might be the noticeable amounts of unseen named entities in the test dataset. On the other hand, BERTurk learns from the entire sentence because of the masked language model. Therefore, this result indicates that BERTurk is a more generalized model than other structures.

5.2.2 CRF Feature Related Experiments

	precision	recall	F1
word.lower()	91.84	78.09	84.41
+ word.postag()	91.44	80.10	85.40
+ word.nounCase()	91.10	81.43	85.99
+ word.properNoun()	91.42	85.36	88.28
+ word.INF()	91.41	86.30	88.78
+ word.isupper()	91.12	86.10	88.54
+ word.isdigit()	91.22	86.15	88.61
+ word.istitle()	91.91	89.23	90.55
+ BOS	91.72	89.34	90.51
+ postag[:2]	91.67	89.29	90.46
+ +1:word.lower()	92.42	90.56	91.48
+ +1:word.istitle()	92.38	90.56	91.46
+ +1:word.isupper()	92.35	90.54	91.43
+ +1:postag	92.38	90.56	91.46
+ +1:postag[:2]	92.40	90.61	91.50
+ -1:word.lower()	92.80	91.12	91.96
+ -1:word.istitle()	92.69	90.94	91.81
+ -1:word.isupper()	92.87	91.07	91.96
+ -1:postag	92.92	91.12	92.01
+ -1:postag[:2]	92.92	91.12	92.01
+ word[-2:]	92.95	91.53	92.24
+ word[-3:]	93.35	91.96	92.65

Table 5: Precision recall and F1 measure in CoNLL metric

In this table, the contribution of the selected features to the scores was analyzed. We have started with word.lower(), which is the surface form of the word as our baseline. A plus(+) sign in the rows stands for adding the feature to the previously used features. For instance, in row 3, we have +word.properNoun() and its corresponding precision, recall, and F1 score. The F1 score in each row is calculated by adding the feature in the row to the features given in the previous rows. In this example, word.lower(), word.postag(), word.nounCase() are the features that form the F1 score of the row. Using morphological features such as POS tag, Noun Case, being proper nouns, and inflections increased the F1 score. Checking all the words which have all their characters uppercase, decreased the score. The reason behind this drop may be the shortage of such samples in the dataset. Adding the feature for being numerical or not, declined the F1 score slightly. While word.istitle() feature raises the F1 score, BOS did not improve the score. This is because word.istitle() methods are always true for the words which are at the beginning of the sentence, which implies that an istitle() feature increased the F1 score. Whereas, +1word.isupper() and +1postag[:2] features slightly decreased the F1 score. Moreover, adding the previous token to the feature sets of CRF raised the score. Conversely, using the word.title() feature and POS tag and last two characters of POS tag dropped to score. Finally, we have examined the effect of using the last two and last three characters of the word. Using the last three characters of a word as a feature, increased the score. However, adding the last two characters as another feature, decreased the results. Since the last three characters carry information of the last two characters, it is not surprising to see no improvement.

5.2.3 BERT vs Error Analysis

- Eco da Carrière de iyi birer entelektüel olmalarının yanında iflah olmaz birer bibliyofil , kitaba tutkuyla bağlı kişiler. Bu açıdan bakıldığında Eco ile Carrière ’ in görüşlerine katılmamak olanaksız.
- TEMSA ve Türk Savunma Sanayisinin lider kuruluşu ASELSAN güçlerini birleştirdi .
- Kendimizi Türkiye ’ nin Sabancı ’ sı olarak adlandırıyoruz . Türkiye ’ nin rekabetçiliğinin de ancak katma değerli üretimle mümkün olduğunu biliyor ve bunun için çalışıyoruz .

Token	True Label	CRF	BERTurk Prediction
Carriere	B-PERSON	B-LOC	B-PERSON
ASELSAN	B-ORG	B-LOC	B-ORGANIZATION
Sabancı	B-ORG	B-PERSON	B-ORG

Table 6: Mispredicted tokens by CRF

On the other hand, it is valuable to see differences between BERTurk with CRF by sentence-by-sentence examination. For instance, CRF tends to predict wrong in non-Turkish words as it can be seen in the first sentence. CRF model is trained with a small Turkish NER dataset, which resulted in accordingly low result in the new test case. As can be seen in results; while the BERTurk score just decreased by 0.06, other models decreased averagely 0.15. CRF and Zemberek which are trained with Turkish NER Dataset more tend to predict non-Turkish entities wrongly. The second sentence which is a good example to see how models perform in tokens that not find in training set shows that CRF and Zemberek tend to predict this tokens to "O". For instance, ASELSAN which is the company in Turkey is not be exampled in Turkish NER Dataset. Therefore, CRF and Zemberek could not encounter ASELSAN through the training process, whereas thanks to pretraining step, BERT has already seen the word 'ASELSAN' and correctly assigned the label. And finally, in the third sentence, it can be said that conditional models tend to predict ambiguous words wrongly. In this case, Sabancı is a company in Turkey; however, it is also surname of the founder of company. In Turkish NER Dataset, Sabancı often exampled with PERSON. Therefore, CRF predicts "Sabancı" as a "PERSON". However, BERTurk which is trained with large corpus and bidirectional contextual algorithm could easily differentiate word according to its context. Consequently, BERT which is a pre-trained transformer model overperforms in extra-ordinary cases such as Eco da Carrière, ASELSAN, and Sabancı. This is because the CRF model could not encounter these words through training. However, BERT is trained with the corpus which is accordingly bigger than the Turkish NER Dataset.

6 Conclusion

In this study, first of all, background linking and wikification tasks of Text Retrieval Conference (TREC) were submitted. Bag of Words, Universal Sentence Encoder and BERT Summarization approaches were used for the background linking task. Stanford NLP Framework, ElasticSearch and Doc2Vec were used for the wikification task. Secondly, a new dataset for Turkish has begun to be collected from the RSS feeds of 13 different online news websites. Finally, Named Entity Recognition (NER) was studied for Turkish language. Using Conditional Random Field (CRF) and BERT, the current state-of-the-art results in Turkish NER were nearly reproduced. The transformer-based model has been observed to outperform other models.

References

Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source nlp framework for turkic languages. *Structure*, *10*, 1–5.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

Aras, G., Makaroglu, D., Demir, S., & Cakir, A. (2020). An evaluation of recent neural sequence tagging models in turkish named entity recognition. *arXiv preprint arXiv:2005.07692*.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., ... others (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

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).

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

Fayoumi, K., & Yeniterzi, R. (2019). Ozu-nlp at trec news 2019: Entity ranking. In *Trec*.

Güngör, O., Güngör, T., & Üsküdarli, S. (2019). The effect of morphology in named entity recognition with sequence tagging. *Natural Language Engineering*, *25*(1), 147–169.

Küçük, D., Arıcı, N., & Küçük, D. (2017). Named entity recognition in turkish: Approaches and issues. In *International conference on applications of natural language to information systems* (pp. 176–181).

Küçük, D., et al. (2009). Named entity recognition experiments on turkish texts. In *International conference on flexible query answering systems* (pp. 524–535).

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

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).

Lu, K., & Fang, H. (2019). Leveraging entities in background document retrieval for news articles. In *Trec*.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

Oflazer, K. (1994). Two-level description of turkish morphology. *Literary and linguistic computing*, *9*(2), 137–148.

Schweter, S. (2020, April). *Berturk - bert models for turkish.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.3770924` doi: 10.5281/zenodo.3770924

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

Sen, M. U., & Yanıkoglu, B. (n.d.). Suder turkce haber derlemlerinin dokuman sınıflandırması document classification of suder turkish news corpora.

Soboroff, I., Huang, S., & Harman, D. (2019). Trec 2019 news track overview. In *Trec.*

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. (2000). *A statistical information extraction system for turkish* (Unpublished doctoral dissertation). Bilkent University.

Tur, G., Hakkani-Tur, D., & Oflazer, K. (2003, 06). A statistical information extraction system for turkish. *Natural Language Engineering*, *9*, 181 - 210. doi: 10.1017/S135132490200284X

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

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … others (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, arXiv–1910.

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