Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 1067 – 1083

doi:10.3906/elk-2108-167

# Event-related microblog retrieval in Turkish

**Cagri TORAMAN***
Aselsan Research Center, Ankara, Turkey

**Received:** 25.08.2021 • **Accepted/Published Online:** 25.12.2021 • **Final Version:** 21.03.2022

**Abstract:** Microblogs, such as tweets, are short messages in which users are able to share any opinion and information. Microblogs are mostly related to real-life events reported in news articles. Finding event-related microblogs is important to analyze online social networks and understand public opinion on events. However, finding such microblogs is a challenging task due to the dynamic nature of microblogs and their limited length. In this study, assuming that news articles are given as queries and microblogs as documents, we find event-related microblogs in Turkish. In order to represent news articles and microblogs, we examine encoding methods, namely traditional bag-of-words and word embeddings provided by BERT and FastText pretrained language models based on deep learning. We find the distance between the encoded news article and microblog to measure text similarity or relatedness between them. We then rank microblogs according to their relatedness to the input query. The experimental results show that (i) BERT-based model outperforms other encoding methods in Turkish, though bag-of-words with Dice similarity has a challenging performance in short text; (ii) news title is successful to represent event as query, and (iii) preprocessing Turkish microblogs has positive impact in bag-of-words and also FastText embeddings, while BERT embeddings are robust to noise in Turkish.

**Key words:** Microblogs, natural language processing, text similarity, text preprocessing, tweets, word embedding

## 1. Introduction

Online social networks are rich sources to share opinions and collaborate with other users. Twitter is an example of online social network that is also known as a microblog platform, where users are able to write any message with a limited length, called tweet. We use the keywords ”microblogs” and ”tweets” interchangeably. People consume online social networks not only to connect each other but also get news and information. A recent study shows that 71% of Twitter users get news from this microblog platform**. Another study estimates that 85% of tweets are related to events that are reported in news articles [22].

Finding event-related microblogs is the base of many social computing applications. For instance, terror events are detected and analyzed to incite fear and anger at various times and locations [5]. Online social networks can be analyzed and simulated by using the microblogs related to specific topics [10]. Another example application that is built upon event-related microblogs is opinion mining and sentiment analysis [25].

Our task is defined as, given an event reported in a news article published at a specific time (called *news event*), finding the most recent but relevant microblogs to the event. The task is also known as *microblog*

*Correspondence: ctoraman@aselsan.com.tr

**Paw Research Center (2019). Report [online]. Website https://www.journalism.org/2019/10/02/americans-are-wary-of-the-role-social-media-sites-play-in-delivering-the-news [accessed 28 October 2021].

*retrieval* or *microblog search*, where documents, or microblogs in this study, are searched according to an ad-hoc query. We use the phrases "microblog retrieval" and "finding event-related microblogs" interchangeably. In this study, the input is a query represented by an event-related news article, and the output is a list of microblogs related to the query. That is, we find microblogs related to events that are reported in news articles. In order to identify and analyze Turkish-specific issues on event-related microblog retrieval, we conduct our experiments on a dataset that consists of Turkish news articles and related tweets.

We examine encoding methods to represent query (news articles) and documents (microblogs) and then to find the distance between the encoded news article and microblog to measure text similarity. Recent developments in deep learning enable more advanced techniques for document encoding that capture better semantics with word embeddings [29]. We utilize pretrained language models based on deep learning, as well as the traditional bag-of-words (BOW) approach. Word embeddings provided by two state-of-the-art language models, namely FastText [6] and BERT [13], are employed. FastText is a noncontextual language model that is based on the Word2Vec embeddings model [27] but considers sub-word embeddings by n-grams. Therefore, FastText would be a good candidate for modeling noisy microblogs. BERT is a contextual language model that is built on bidirectional contextual representations of words by considering word positions and order with the Transformer model [40]. We think that BERT can leverage contextual semantics of short text.

The input of event-related microblog retrieval is an ad-hoc query written in natural language text. Since we represent query with news articles, we examine different parts of news article to write a query. We utilize news article's full text, title, and snippet. In addition to query length, we also analyze the effect of document (tweet) length on the model performances.

Text preprocessing is an important phase in microblog retrieval, since microblogs are mostly written in a noisy and slang language. Word stemming aims to find word stems to reduce noise by eliminating multiple versions of the same word. Stop words are frequently observed words across many documents. Removing stop words would reduce noise in the dataset. We apply both stemming and removing stop words, as well as text cleaning.

The contributions of this study are the following three research questions that focus on event-related microblog retrieval, specific to Turkish microblogs.

**RQ1, Encoding:** Which encoding approach (based on traditional bag-of-words or word embeddings provided by pretrained language models) performs better effectiveness for the task of document encoding?

**RQ2, Query selection and text length:** Which part of news articles (full text, title, or snippet) performs better effectiveness for the task of query selection? And what is the effect of text length (query and tweet lengths) on the model performance?

**RQ3, Text preprocessing:** What is the effect of text preprocessing, including stemming and removing stop words, for the task of Turkish microblog retrieval?

The rest of the study is organized as follows. In the following section, we give a summary of related work for event-related microblogs. We explain how we find event-related microblogs in Turkish in Section 3. We, then, present the experimental setup and results in Section 4. We conclude the study in the last section.

## 2. Related work

In this section, we divide the related studies into five categories: (i) Event detection in microblogs, (ii) linking news and microblogs, (iii) microblog retrieval, (iv) deep learning for microblog retrieval, and (v) related efforts

in Turkish.

### 2.1. Event detection in microblogs

The aim of event detection is to classify or cluster event-related microblogs according to predefined labels. The differences from microblog retrieval are that (i) the input of event detection is not necessarily an ad-hoc query, (ii) event detection does not rank the related messages, and (iii) sometimes event detection refers to finding event labels in a microblog collection, rather than finding event-related messages. Another related domain is topic tracking, in which new events or topics are detected among temporally ordered stories of document streams, such as news feed [9].

Event detection is mostly accomplished by exploiting text content [3]. Rather than using text alone, extending content by linking it to semantic Web is an alternative solution for detecting events. For instance, there are efforts to enhance the content of microblogs by linking to other sources such as Wikipedia [26].

### 2.2. Linking news and microblogs

In this study, we narrow the problem to the connection between news events and microblogs. The task of event detection is sometimes called as linking news and microblogs, since news articles are used as the source of events.

TwitterStand [35] filters tweets related to breaking news, based on news-source URLs given in tweets. Abel et al. [1] exploit URLs to detect news sources, and find entities with relations in both news and tweets. Guo et al. [18] propose a graph-based approach to model text-to-text correlations for the Linking-Tweets-to-News task. Wang et al. [41] propose an entity-aware event detection model for multiple aspects of events to link with tweets. Tsagkias et al. [39] study the problem in the opposite way, and link news to various platforms; such as Twitter, Wikipedia, and blog pages.

### 2.3. Microblog retrieval

Our task is an ad-hoc search in microblogs, where query is represented by a news event. TREC Microblog Tracks [30, 36] address a similar ad-hoc search task in which the goal is to find most relevant tweets sorted by time. The information need is represented by a short set of keywords that are not necessarily related to real-time news events, whereas we focus on microblogs related to events that are reported in news articles. Another similar task is search result diversification for microblog retrieval in which the system aims to list tweets from as many subtopics as possible with high effectiveness [28].

Microblog retrieval includes a set of subtasks, including query processing, document indexing, matching queries with documents, and ranking the result list. In this study, we focus on all subtasks, except document indexing. For query processing, we use news article's full text, title, and snippet. For document matching and ranking, we encode documents and queries based on two approaches, namely using bag-of-words and word embeddings and then rank the documents according to the Cosine similarity between the encoded vectors.

### 2.4. Deep learning for microblog retrieval

With the recent developments in Natural Language Processing, the semantics of individual words can be captured by static (non-contextual) and dynamic (contextual) word embeddings. Static word embeddings are context-independent, i.e. the same word would have the same embeddings vector without considering its different meanings. On the other hand, dynamic word embeddings are proposed to capture the different

meanings of words. In this study, we use a static model (FastText [6]), and a dynamic one (BERT [13]). There are efforts to employ BERT embeddings to re-rank microblog retrieval results for the English language [43].

Neural information retrieval is a recent research area that provides effective solutions in retrieval systems [29]. Word embeddings are utilized for various subtasks in information retrieval; such as term weighting [45], query representation [44], and measuring similarities in short-texts [20]. However, the deep learning-based studies are limited in microblog retrieval. Such studies mostly employ non-contextual word embeddings, such as Word2Vec [27] and GloVe [33]. For instance, Word2Vec embeddings are employed with the Cosine similarity to score the similarity between tweets and queries that represent post-disaster needs of resources [4].

Word embedding models are trained with the random initialization of model parameters. However, a recent deep learning technique, called transfer learning [32], does not require pre-training of model parameters on a specific task. In other words, transfer learning enable to transfer advanced word representations to other domains. This adaptation is achieved by fine-tuning the existing pre-trained language model with the given training set. Our objective in this study is not a classification task, thereby does not require pre-training or fine-tuning. We measure the similarity between a query and a tweet vector that is encoded with pre-trained language models.

### 2.5. Related efforts in Turkish microblogs

Event detection is studied in Turkish microblogs with the help of clustering semantically related hashtags [31]. Turkish tweets are analyzed to capture real-time events with their location and time [14]. Neural feature extraction is applied to improve the performance of event detection in Turkish tweets [15]. Linking tweets to news is studied by measuring document similarity in bag-of-words [21]. Event-related tweets are classified by using URLs that link to news sources [12]. Pre-trained language models are also studied to detect topics in Turkish microblogs [34]. However, these studies do not target the task of event-related microblog retrieval.

Although information retrieval is studied for ad-hoc queries in Turkish document collections [7, 8], there are limited studies in terms of event-related microblog retrieval in Turkish. Tweets are searched with respect to news events by using inverted index and boolean search [37]. To the best of our knowledge, this is the first study on event-related microblog retrieval in Turkish that examines query and tweet encoding methods, including FastText and BERT, as well as query selection and text preprocessing.

## 3. Finding event-related microblogs in Turkish

We explain how we find event-related microblogs in this section. Figure 1 shows an illustration of the phases that we focus in event-related microblog retrieval in this study. The main phases are query selection, text preprocessing, query and tweet encoding, and lastly similarity ranking.

### 3.1. Query selection

Given an input news article that reports a real-life event, we first form a query by using one of the three parts of news articles: News article's full text, title, and snippet. We extract full text body of news articles. Since full text of news articles are too long, we select news title and snippet as well. News title is mostly a sentence that describes the event, while news snippet is a short summary of the event. Note that documents are microblogs (tweets) with short text length (at most 280 characters).

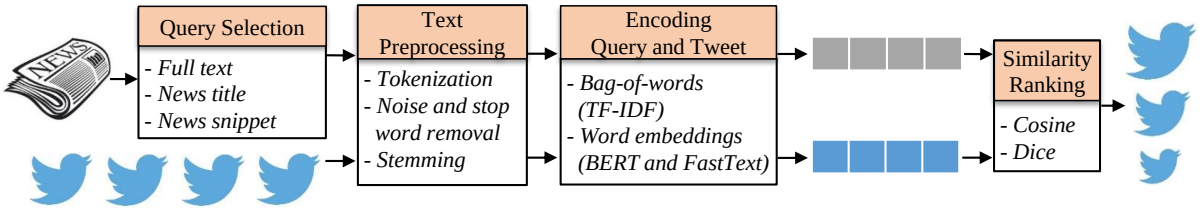


**Figure 1**. An illustration of the phases for finding event-related microblogs in this study.

### 3.2. Text preprocessing

After query is selected, the content of query and tweets are preprocessed, which is an important phase due to unstructured and noisy language used in microblogs. We apply the following text preprocessing steps.

(i) *Tokenization*: For the bag-of-words encoding, we tokenize text by dividing by space and punctuation marks. For the word embedding-based encoding, we use the inner tokenizers provided by FastText and BERT language models. FastText uses the ICU (International Components for Unicode) tokenizer for Turkish, which determines default word segmentation boundaries based on Unicode*. BERT uses the WordPiece approach [42] that masks n-gram characters to find common overlapping character sequences. Since FastText and BERT employ sub-word tokenization, an out-of-vocabulary word can still have a vector representation in the word-embeddings space, such that its sub-word embeddings are summed up by the algorithm.

(ii) *Noise cleaning*: We ignore the symbols that users express their emotions (emoticons) and usernames that are included in tweets (mentions). We remove hashtags starting with the hash symbol that identify messages on a particular topic, since hashtags have no space among words (i.e. we do not normalize hashtags). We remove invalid characters and tokens shorter than three characters. We also divide the words attached to each other due to missing whitespace (e.g. "#First#Second" is converted to "#First #Second").

(iii) *Stop word removal*: We remove frequently used words, called stop word removal. We use the stop word list that is an extended version of the list given in [8].

(iv) *Stemming*: Finding word stems can reduce the potential noise in agglutinative languages, such as Turkish, due to the fact that many suffixes or other morphemes can be added to the base of words. We apply the F$k$ stemmer [38], which considers the first $k$ characters of words and ignores the remaining ones. We also apply the Zemberek lemmatizer [2], to compare the performance of stemmer with a lemmatization method. Note that both stemming and lemmatization find the root of a word, but lemmatization considers the canonical or dictionary form of a word. For instance, a lemmatizer finds the root of "kitabı" (translated as "his/her book") as "kitap" (translated as "book"), whereas a stemmer finds "kitab". The dictionary form of a word can increase the chance of detecting pretrained word embeddings for that word in the given language model.

*Unicode (2020). Unicode Text Segmentation [online]. Website http://www.unicode.org/reports/tr29 [accessed 28 October 2021].

### 3.3. Encoding query and tweets

Preprocessed query and tweets are encoded into the vectors that are represented in the bag-of-words (BOW) and word embedding-based models.

#### 3.3.1. BOW-based encoding

Bag-of-words is a conventional model that considers no word order and dependencies. BOW is based on vector space model, where each document is represented in a fixed length of vectors. Each vector consists of identifiers for terms in documents. We use TF-IDF (Term Frequency-Inverse Document Frequency) term weighting scores. For a given term $t$ and document $d$, TF-IDF is calculated by multiplication of term frequency and inverse document frequency, as follows.

$$\text{TF-IDF}(t,d) = tf(t,d) \times idf(t) \tag{1}$$

Term frequency (tf) measures the importance of a term in a document. Term frequency of a term, $t$, in a document, $d$, is calculated as the number of occurrences of the term in the document ($n_{t,d}$) divided by the number of terms in the document ($n_d$), as follows.

$$\text{tf}(t,d) = \frac{n_{t,d}}{n_d} \tag{2}$$

Inverse document frequency (idf) measures the importance of a term in a document collection. Inverse document frequency of a term, $t$, is calculated as the total number of documents in the collection ($M$) divided by the number of documents that term appears ($M_t$), as follows.

$$\text{idf}(t) = \log \frac{M}{M_t} \tag{3}$$

#### 3.3.2. Word embedding-based encoding

For the embeddings-based encoding of query and tweets, we first obtain word embeddings for each token from a pre-trained language model. We do not train FastText and BERT, but use existing pre-trained models [6, 13]. FastText word embeddings are trained on Wikipedia articles and web pages including news articles. On the other hand, BERT word embeddings are trained on Wikipedia articles and books. Assume that $w \epsilon \mathbb{R}^r$ is a word embedding vector, and $r$ is the size of embedding dimension ($r$ is 300 for FastText embeddings and 768 for BERT embeddings in the experiments of this study). In order to encode query and tweets with word embeddings, we use the following methods.

1. We calculate average of word embeddings (AWE) of individual tokens as follows.

$$awe(S) = \frac{\sum_{w_i \epsilon S} w_i}{|S|} \tag{4}$$

where $S$ is either a query or tweet, $|S|$ is the length of a query or tweet in terms of the number of tokens, and $w \epsilon \mathbb{R}^r$ is a word embedding vector. The sum operation is element-wise sum of vectors.

2. We calculate average of word embeddings (AWE) of individual terms with L2 normalization. Each word embedding vector is normalized by L2-norm, which is the length of a vector in the Euclidean space. Average of word embeddings that have positive L2-norm scores are calculated as in (4).

3. For only BERT, we also use the hidden sentence embeddings (CLS) [13] that is embedded in the latest layer of BERT to represent the semantics of a given sentence (in our case query or tweet) in a fixed length vector.

### 3.4. Similarity ranking

After encoding query and tweets into two families of vectors (BOW or word embedding-based), we calculate the Cosine similarity score between the encoded vectors of a query and tweet. Given the vector length $L$, the Cosine similarity score between a query, $q \epsilon \mathbb{R}^L$, and tweet, $t \epsilon \mathbb{R}^L$, is measured as follows.

$$cos(q,t) = \frac{\sum_{i=1}^{L} q_i t_i}{\sqrt{\sum_{i=1}^{L} (q_i)^2} \sqrt{\sum_{i=1}^{L} (t_i)^2}} \tag{5}$$

The Cosine similarity score might not work properly when query and tweet vectors are sparse, which is likely to observe in microblogs. We assume that descriptive words are mostly used for news events in microblogs. Therefore, we expect to match overlapping descriptive words by the Dice similarity score between query and tweet, as follows.

$$dice(q,t) = \frac{2n_p}{n_q + n_t} \tag{6}$$

where $n_p$ is the number of overlapping tokens between query and microblog, $n_q$ is the length of query in terms of the number of tokens, $n_t$ is the length of tweet. We use TF (Term Frequency) term weighting while calculating the Dice similarity score.

## 4. Experiments

In this section, we first explain our experimental setup; including the dataset, metrics, and methodology that we use in the experiments. We, then, report and discuss the experimental results.

### 4.1. Dataset

We use a dataset that includes Turkish news articles and related microblogs (tweets), called BilTweetNews-2017*. The dataset contains tweets related to six major events from Turkish news sources between May 4, 2015 and Jan 8, 2017. Since our approach for finding event-related microblogs has no need for training, we use the whole dataset in evaluation. In this section, we describe the process of collecting tweets and news, annotation by users, and lastly validating of a reliable ground truth statistically.

*The download link for the dataset is https://github.com/BilkentInformationRetrievalGroup/BilTweetNews2017

#### 4.1.1. Collecting data

We acquire all tweets published by Twitter API in 2015 and 2016, and select six major events occurred in Turkey in the same period. Tweets to be annotated are pooled from a time window of five-day length, starting from the beginning of each event.

Related candidate tweets are chosen by searching some keywords that we predetermine according to the content of events. Unrelated candidates are chosen by searching the same keywords, outside of the time window, but among tweets far away in timeline. These tweets are expected to be unrelated to the events, but having the same keywords would make difficult to distinguish related and unrelated ones for simple algorithms. For example, we select tweets that contain the keyword, Madonna, but not the ones related to the event of TV magazine host confusing Madonna. For each six major events, 100 related-candidate and 60 unrelated-candidate tweets are selected. Lastly, we randomly select 40 tweets that are potentially not related at all, 5 of them are removed due to detecting near-duplicates later. The dataset has 995 tweets in total. The average tweet length is 12.9 words and 105.1 characters when the text is split to words by whitespaces. After preprocessing, the average tweet length is 14.1 words and 104.0 characters, since we clean and normalize text as described in Section 3.2. After preprocessing, we create an inverted index with 3898 unique tokens for representing the bag-of-words model.

#### 4.1.2. Data annotation

All tweets are assigned to 17 annotators, mostly graduate students and a few faculty members. We develop an annotation web-page specifically designed for our task. Annotators are assigned to the same 10 tasks, each including around 100 tweets. Annotators can observe the status of their tasks with the start and end date. They are also able to save and continue a task later by saving the labels.

Users label the relatedness of a tweet to six events, represented by news articles. Users can also assign tweet to none of the events. The content of news articles can be accessed to comprehend events. Table 1 lists the event dates, descriptions, and their number of instances when majority voting is applied. There are no cases that contains multiple labels for events, i.e., users mostly agree on a specific event.

**Table 1**. The details of the events in the dataset.

| **Event Date** | **Event Description** | **Annotated Tweets** |
|---|---|---|
| - | Not related to any news topic | 590 |
| May 25, 2015 | One of the popular football clubs in Turkey, Galatasaray, wins the 2015 Turkish Super League. | 46 |
| Sep 6, 2015 | A terrorist group, called PKK, attacked to soldiers in Dağlıca, a village in southeastern Turkey. | 103 |
| Oct 7, 2015 | A Turkish scientist, Aziz Sancar, won the 2015 Nobel Chemistry prize with his studies on DNA repair. | 89 |
| May 27, 2016 | A local football club of Alanya promoted to the Turkish Super League for the first time in their history. | 41 |
| Jun 17, 2016 | A traditional anthem that is mostly played by secularists in Turkey, called the 10th Year Anthem, was forbidden in schools by the director of national education in the Black Sea province of Bolu. | 38 |
| Oct 17, 2016 | A magazine programmer confused that *Madonna in a Fur Coat*, a book written in 1943 by a Turkish celebrated writer, Sabahattin Ali, was about popstar Madonna's life. The book tells a story between a Turkish student and German singer after the World War I. | 88 |

#### 4.1.3. Interagreement and outlier detection

Inter-agreement of annotators is measured by Fleiss' kappa. [16]. Fleiss' kappa scores is 0.78 for the event annotation, meaning substantial agreement according to the interpretation of Landis and Koch [23].

For outlier detection, we first calculate Fleiss' kappa without each annotator separately. The scores obtained without individual annotators are given to the box-plot method, which gives outliers in a given distribution. If box-plot gives any outlier, it means that removing the corresponding annotator provides a significant difference in consistency, hence becoming an outlier for our annotations. However, box-plots indicate no outliers.

### 4.2. Metrics

We report the effectiveness performance with mean average precision (MAP) and Precision at top 30 (P@30), which are used in the TREC Microblog tracks [30, 36]. Given a query set, $Q$, MAP is the mean of the average precision (AP) scores of each query, $q$, as follows.

$$MAP = \frac{1}{|Q|} \sum_{q \epsilon Q} AP_q \tag{7}$$

Given a search result list, $L$, AP is the average of the precision scores at each ranking position, $k$, as follows.

$$AP = \frac{1}{r} \sum_{k=1}^{|L|} Precision@k \tag{8}$$

Precision at a ranking position, $k$, is calculated as follows.

$$Precision@k = \frac{\text{number of relevant tweets at rank k}}{\text{number of all tweets at rank k}} \tag{9}$$

Using the effectiveness scores of queries as independent observations, the possible improvement in a single pairwise comparison between two effectiveness scores is statistically validated using the one-tailed paired t-test at 95% interval, where we are interested in the difference over a baseline method.

### 4.3. Experimental methodology

We employ word embeddings that are provided by BERT [13] and FastText [6]. We use BERT's multilingual cased pre-trained model on Wikipedia articles [13] and easy-bert**; and FastText's word embeddings pre-trained on Turkish Wikipedia articles [17] and FastText4j*. We also use Weka [19] for implementing bag-of-words and similarity measurements. In the experiments, we compare the following encoding methods.

(i) **bowCOS**: A baseline model that employs bag-of-words with TF-IDF and the Cosine similarity ranking.

(ii) **bowDICE**: A baseline model that employs bag-of-words with TF and the Dice similarity ranking.

**Github (2020). easy-bert [online]. Website https://github.com/robrua/easy-bert [accessed 28 October 2021].

*Github (2020). fastText4j [online]. Website https://github.com/linkfluence/fastText4j [accessed 28 October 2021].

(iii) **embBERT**: Word embedding-based models with BERT embeddings and the Cosine similarity ranking. Three versions are examined: average word embeddings (AWE), average word embeddings with L2 normalization (AWE-L2), and sentence embeddings (CLS) that are obtained from the last layer of BERT.

(iv) **embFAST**: Word embedding-based models with FastText embeddings and the Cosine similarity ranking. Two versions are examined: Average word embeddings (AWE), and average word embeddings with L2 normalization (AWE-L2).

Since we rank all tweets, we do not set a threshold for the Cosine and Dice similarity measures to calculate the MAP scores. For the P@30 score, we consider the top-30 tweets in the ranking list.

We also analyze the effects of query selection and text preprocessing. For query selection, we compare the following selection methods having varying text lengths.

(i) **Full Text**: A baseline approach that uses the whole text of a news article as query.

(ii) **Title**: Using news title as query, which consists of a single sentence.

(iii) **Snippet**: Using news snippet as query, which consists of a few sentences.

In addition to query length, we also analyze document (tweet) length for the model performance. We consider tweet length as the number of words, and rerun all models for different tweet subsets of varying length thresholds from 5 to 30 words.

For text preprocessing, we compare the following preprocessing methods.

(i) **None**: A baseline approach that applies no text preprocessing at all.

(ii) **Zemberek**: Using the preprocessing steps that are listed in Section 3.2 with the Zemberek lemmatizer and stop word removal.

(iii) **F5**: Using the preprocessing steps that are listed in Section 3.2, with the F5 stemmer and stop word removal.

### 4.4. Experimental results

In this section, we report our experimental results with respect to the research questions of this study; namely the analysis of encoding, query selection and text length, and text preprocessing.

#### 4.4.1. Encoding query and microblogs

The comparison of the bag-of-words and word embedding-based encoding models are given in terms of MAP and P@30 in Table 2. In this experiment, we use news title as query and apply no preprocessing. We observe that average word embeddings of BERT with L2 normalization, *embBERT (AWE-L2)*, outperforms the baselines and other embedding-based methods in terms of MAP. We argue that BERT's contextual embeddings can capture semantics of short text successfully without any pretraining or fine-tuning.

The top-3 ranking of the encoding methods in terms of MAP in decreasing order is (i) *embBERT (AWE-L2)*, (ii) *bowDICE*, and (iii) *embFAST (AWE-L2)*. The Dice similarity with bag-of-words (*bowDICE*) has the second highest MAP score. We argue that overlapping important keywords that are used in short texts (news titles and microblogs) can be captured with the Dice similarity. The Cosine similarity performs worse than other methods in MAP, as we expect due to the sparseness problem, while *bowCOS* has the highest P@30 score.

**Table 2**. The comparison of the methods for encoding query and microblogs (**RQ1**). The baseline methods are based on bag-of-words (*bowCOS* and *bowDICE*). The bold score is the highest.

| **Encoding** | **MAP** | **P@30** |
|---|---|---|
| *bowCOS* | 0.499 | **0.411** |
| *bowDICE* | 0.524 | 0.389 |
| embBERT (CLS) | 0.304 | 0.189 |
| embBERT (AWE) | 0.509 | 0.344 |
| embBERT (AWE-L2) | **0.538** | 0.350 |
| embFAST (AWE) | 0.492 | 0.378 |
| embFAST (AWE-L2) | 0.514 | 0.367 |

BERT CLS embeddings perform poor, which can be attributed to the fact that, as its name suggests, CLS embeddings are proposed for fine-tuning classification tasks [13]. Average word embeddings (AWE) have higher effectiveness for BERT when L2 normalization is used. This observation is similar for FastText embeddings in terms of MAP. Note that we employ pretrained word embeddings in this study. We argue that state-of-the-art language models, such as BERT and FastText, can be fine-tuned to improve the effectiveness performance of event-related Turkish microblogs.

The answer to the first research question (which encoding approach performs better effectiveness for the task of document encoding?) could be pre-trained word embeddings by using the BERT language model and L2 normalization, based on the MAP scores.

#### 4.4.2. Query selection and text length

The comparison of the methods for query selection are given in terms of MAP and P@30 in Table 3. In this experiment, we apply no preprocessing. Note that the ranking in decreasing order according to text length is (i) full text, (ii) snippet, and (iii) title. That is, title is the shortest text, whereas full text is the longest text in terms of the number of both words and characters.

We observe that, for all encoding methods, using news title as query performs the highest effectiveness in terms of MAP, compared to full text and snippet. The results are very similar in terms of P@30; the only exception is that news snippet performs the highest in terms of P@30 for *bowDICE*.

When the query is relatively longer, e.g., when full text is selected as query, *bowCOS* performs the highest effectiveness. The reason would be that the Cosine similarity would perform poor in sparse vectors, which is not likely to be seen in long text. For the word embedding-based methods, the effectiveness performance is deteriorated as text length gets longer. We argue that averaging word embeddings of long texts would cause noise and, thereby, lose semantics. Note that this effectiveness gap between short and long texts is larger in BERT, compared to FastText (the MAP difference between title and full text is 0.210 in BERT, and 0.159 in FastText).

We also analyze the effect of text length on the model performance in terms of document length (tweet length). Figure 2 shows the model performance in terms of MAP and P@30 when tweet length (the number of words) is limited to a threshold value. For instance, the scores reported for 20 at x-axis are measured by using the tweets having less or equal to 20 words. Note that the longest tweet length is not longer than 30 in our dataset; therefore, we plot up to 30 words. We observe that the MAP and P@30 scores of all models

**Table 3**. The comparison of the methods for query selection (**RQ2**). The bold score is the highest. (*) and (**) indicate approaching ($p < .10$) and highly ($p < .05$) statistically significant improvements, respectively, compared to the baseline (*Full Text*).

| Encoding | Query | MAP | P@30 |
|---|---|---|---|
| bowCOS | *Full Text* | 0.473 | 0.383 |
| | Title | **0.499** | **0.411** |
| | Snippet | 0.443 | 0.350 |
| bowDICE | *Full Text* | 0.447 | 0.344 |
| | Title | **0.524** | 0.389 |
| | Snippet | 0.458 | **0.411** |
| embBERT | *Full Text* | 0.328 | 0.222 |
| | Title | **0.538**** | **0.350**** |
| | Snippet | 0.391** | 0.250** |
| embFAST | *Full Text* | 0.355 | 0.289 |
| | Title | **0.514*** | **0.367** |
| | Snippet | 0.428 | 0.361* |

improve as tweet length increases. The performances are converged after 25 words. All models have similar performance scores when tweets have short length. The performance gap among the models increase for longer tweets. Another observation is that *embFAST* outperforms *bowDICE* for the tweets having less than 20-25 words, but *bowDICE* gets better performance than *embFAST* for longer tweets.

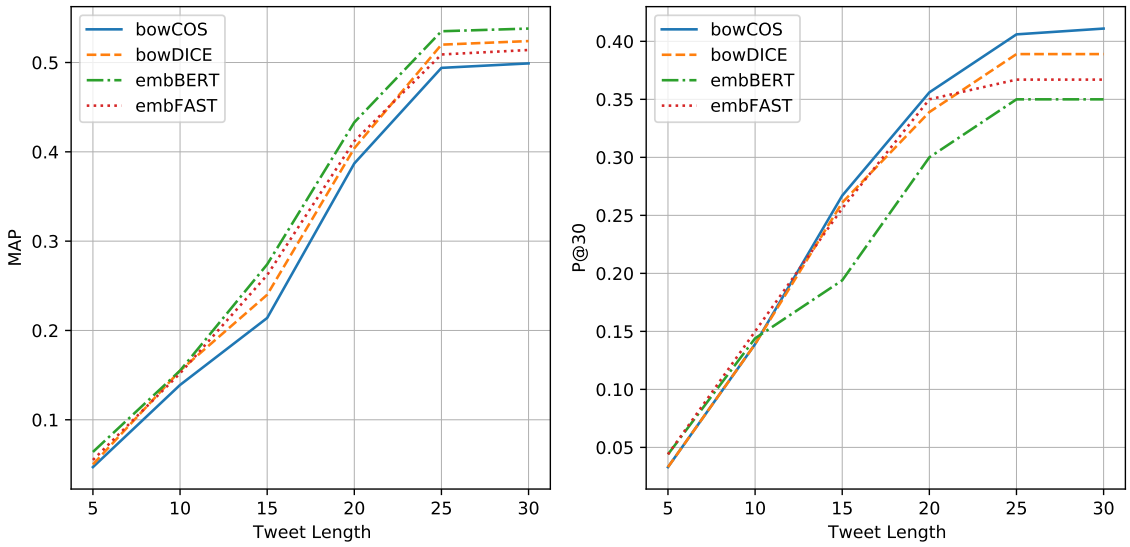


**Figure 2**. Effect of the tweet length (the number of words) on the model performance.

As an answer to the second research question, our experimental results show that the title part of news articles performs better effectiveness for the task of query selection. In other terms, short queries outperforms

longer ones. In addition, we observe that the model performance also depends on document (tweet) length, where the best scores are obtained for the tweets with 25 or more words.

#### 4.4.3. Text preprocessing

The effect of text preprocessing is reported in terms of MAP and P@30 in Table 4. In this experiment, we use news title as query. Note that our text preprocessing steps, listed in Section 3.2, include noise and stop word removal, as well as stemming words. We report the results that employ the Zemberek lemmatizer and F5 stemmer.

**Table 4**. The effect of text preprocessing (**RQ3**). The bold score is the highest. (*) and (**) indicate approaching ($p < .10$) and highly ($p < .05$) statistically significant improvements, respectively, compared to the baseline (*None*).

| Encoding | Preprocess | MAP | P@30 |
|---|---|---|---|
| bowCOS | *None* | 0.499 | 0.411 |
| | Zemberek | 0.516 | 0.406 |
| | F5 | **0.560*** | **0.433** |
| bowDICE | *None* | 0.524 | 0.389 |
| | Zemberek | 0.588* | 0.406 |
| | F5 | **0.624**** | **0.450*** |
| embBERT | *None* | 0.538 | **0.350** |
| | Zemberek | **0.539** | 0.322 |
| | F5 | 0.509 | 0.317 |
| embFAST | *None* | 0.514 | 0.367 |
| | Zemberek | 0.562 | 0.378 |
| | F5 | **0.568*** | **0.400*** |

We observe that applying text preprocessing improves effectiveness for BOW-based models (*bowCOS* and *bowDICE*). The reason that text preprocessing improves effectiveness for BOW-based models would be that Turkish is an agglutinative language that has many morphemes added to the end of words, and BOW processes terms independently. On the other hand, we observe that the performance of BERT embeddings is not improved by text preprocessing. The reason that text preprocessing does not improve effectiveness for BERT would be that the input words are segmented into subwords inherently in BERT by using the WordPiece approach [42]. However, we observe consistent improvement for FastText embeddings that divides words into subwords as well. We argue that FastText's sub-word approach is not successful in finding Turkish stems, while BERT's WordPiece approach is successful in Turkish. Note that FastText averages the n-gram sub-words of a word to represent central word embedding, while BERT masks n-gram characters in a word to find common overlapping character sequences, so that unknown or out-of-vocabulary words can be represented with subword embeddings.

In terms of comparing stemming and lemmatization, we observe that both the Zemberek lemmatizer and F5 stemmer improve effectiveness for BOW-based models, except Zemberek in P@30 for *bowCOS*. Although lemmatization finds dictionary form of words, the F5 stemmer has higher effectiveness scores, compared to the Zemberek lemmatizer, in event-related microblog retrieval. This observation is also seen in Turkish ad-hoc information retrieval as well [7].

The answer to the last research question could be that text preprocessing improves performance for all models except BERT, based on our experimental results.

## 5. Conclusion and future work

In this study, we aim to find event-related Turkish microblogs, where news articles are given as queries, and tweets as documents. We consider three aspects of event-related microblog retrieval. First, we compare two families of encoding methods, namely traditional bag-of-words and pre-trained word embeddings that are provided by BERT and FastText language models based on deep learning. Second, we compare the effectiveness of using news article's full text, title, and snippet for representing event as query. Lastly, we examine the effects of applying text preprocessing.

We conduct experiments in a Turkish microblog dataset to show the effectiveness performance of the aforementioned methods for the Turkish language. The experimental results show that BERT-based model outperforms other encoding methods in Turkish. Traditional bag-of-words model with the Dice similarity measurement has a challenging performance in short text as well. In order to represent events as queries, we observe that using news title as query outperforms that of full text and snippet. Lastly, we show that preprocessing Turkish microblogs has positive impact in bag-of-words and also FastText embeddings. However, BERT embeddings are robust to noise in Turkish with the help of its tokenization approach, i.e., text preprocessing is not an essential step for BERT embeddings in Turkish. We note that the performance could increase by using language models that are pretrained on Turkish microblogs, since social media text can be short and noisy. Although there are efforts* for pre-training BERT for Turkish text, there is still lack of pre-trained language models for Turkish microblogs.

In future work, we plan to reflect more results to understand in what circumstances bag-of-words and word embeddings differ in event-related microblog retrieval in Turkish. Other methods to represent sentence embeddings, such as InferSent [11], can be examined to improve effectiveness. BERT and FastText can be fine-tuned for a classification task that would label whether tweet is relevant to query. In case of having huge amount of microblogs or tweets, one can discuss to ignore ranking of some of those tweets for the sake of efficiency. There are some approaches to ignore such tweets in the creation of the inverted index structure [46].

## References

[1] Abel F, Gao Q, Houben G, Tao K. Semantic enrichment of Twitter posts for user profile construction on the social web. In: The Semanic Web: Research and Applications - 8th Extended Semantic Web Conference ESWC 2011; Crete, Greece; 2011. pp. 375–389. doi:10.1007/978-3-642-21064-8_26

[2] Akın AA, Akın MD. Zemberek, an open source NLP framework for Turkic languages. Structure, 2007; 10: 1–5.

[3] Atefeh F, Khreich W. A survey of techniques for event detection in Twitter. Computational Intelligence, 2015; 31 (1): 132–164. doi:10.1111/coin.12017

[4] Basu M, Ghosh K, Das S, Dey R, Bandyopadhyay S et al. Identifying post-disaster resource needs and availabilities from microblogs. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining; Sydney, Australia; 2017. pp. 427–430. doi:10.1145/3110025.3110036

[5] Baucum M, Cui J, John RS. Temporal and geospatial gradients of fear and anger in social media responses to terrorism. ACM Transactions on Social Computing 2020; 2 (4): 1–16. doi:10.1145/3363565

*BERTurk (2021) [online]. Website https://github.com/stefan-it/turkish-bert [accessed 28 October 2021].

[6] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 2017; 5: 135–146. doi:10.1162/tacl_a_00051

[7] Can F, Kocberber S, Balcik E, Kaynak C, Ocalan HC et al. First large-scale information retrieval experiments on Turkish texts. In: Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval; Seattle, Washington, USA; 2006. pp. 627–628. doi:10.1145/1148170.1148288

[8] Can F, Kocberber S, Balcik E, Kaynak C, Ocalan HC et al. Information retrieval on Turkish texts. Journal of the American Society for Information Science and Technology 2008; 59 (3): 407–421. doi:10.1002/asi.20750

[9] Can F, Kocberber S, Baglioglu O, Kardas S, Ocalan HC et al. New event detection and topic tracking in Turkish. Journal of the American Society for Information Science and Technology 2010; 61 (4): 802–819. doi:10.1002/asi.21264

[10] Chung W, Toraman C, Huang Y, Vora M, Liu J. A deep learning approach to modeling temporal social networks on Reddit. In: IEEE International Conference on Intelligence and Security Informatics; Shenzhen, China; 2019. pp. 68–73. doi:10.1109/ISI.2019.8823399

[11] Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A. Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing; Copenhagen, Denmark; 2017. pp. 670–680. doi:10.18653/v1/D17-1070

[12] Demirsoz O, Ozcan R. Classification of news-related tweets. Journal of Information Science, 2017, 43 (4): 509–524. doi:10.1177/0165551516653082

[13] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); Minneapolis, Minnesota; 2019. pp. 4171–4186. doi:10.18653/v1/N19-1423

[14] Erdoğan AE, Yilmaz T, Sert OC, Akyüz M, Özyer T et al. From social media analysis to ubiquitous event monitoring: The case of Turkish tweets. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; Sydney, Australia; 2017. pp. 1088–1095. doi:10.1145/3110025.3120986

[15] Ertugrul AM, Velioglu B, Karagoz P. Word embedding based event detection on social media. In: 2017 International Conference on Hybrid Artificial Intelligence Systems; La Rioja, Spain; 2017. pp. 3–14. doi:10.1007/978-3-319-59650-1_1

[16] Fleiss JL. Measuring nominal scale agreement among many raters. Psychological Bulletin 1971; 76 (5): 378–382.

[17] Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T. Learning word vectors for 157 languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation; Miyazaki, Japan; 2018. pp. 3483–3487.

[18] Guo W, Li H, Ji H, Diab M. Linking tweets to news: A framework to enrich short text data in social media. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Sofia, Bulgaria; 2013. pp. 239–249.

[19] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P et al. The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter 2009; 11 (1): 10–18. doi:10.1145/1656274.1656278

[20] Kenter T, Rijke de M. Short text similarity with word embeddings. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management; Melbourne, Australia; 2015. pp. 1411–1420. doi:10.1145/2806416.2806475

[21] Kulcu S, Dogdu E. A scalable approach for sentiment analysis of Turkish tweets and linking tweets to news. In: 2016 IEEE Tenth International Conference on Semantic Computing; Laguna Hills, CA, USA; 2016. pp. 471–476. doi:10.1109/ICSC.2016.66

[22] Kwak H, Lee C, Park H, Moon S. What is Twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web; Raleigh, North Carolina, USA; 2010. pp. 591–600. doi:10.1145/1772690.1772751

[23] Landis JR, Koch GG. The measurement of observer agreement for categorical data. Biometrics, 1977; 33 (1).

[24] Liew JSY, Turtle HR, Liddy ED. EmoTweet-28: A fine-grained emotion corpus for sentiment analysis. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation; 2016. pp. 1149–1156.

[25] Liu B, Zhang L. A survey of opinion mining and sentiment analysis. In: Aggarwal C, Zhai C (editors), Mining Text Data. Boston, MA, USA: Springer, 2012, pp. 415–463. doi:10.1007/978-1-4614-3223-4_13.

[26] Meij E, Weerkamp W, Rijke de M. Adding semantics to microblog posts. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining; Seattle, Washington, USA; 2012. pp. 563–572. doi:10.1145/2124295.2124364

[27] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2; Lake Tahoe, Nevada; 2013. pp. 3111–3119.

[28] Onal KD, Altingovde IS, Karagoz P. Utilizing word embeddings for result diversification in tweet search. In: 11th Asia Information Retrieval Societies Conference; Brisbane, QLD, Australia; 2015., 2015 pp. 366–378. doi:10.1007/978-3-319-28940-3_29

[29] Onal KD, Zhang Y, Altingovde IS, Rahman MM, Karagoz P et al. Neural information retrieval: At the end of the early years. Information Retrieval 2018; 21 (2–3): 111–182. doi:10.1007/s10791-017-9321-y

[30] Ounis I, Macdonald C, Lin J, Soboroff I. Overview of the TREC 2011 microblog track. In: Proceedings of the Twentieth Text REtrieval Conference (TREC 2011); Gaithersburg, Maryland, USA; 2011.

[31] Ozdikis O, Senkul P, Oguztuzun H. Semantic expansion of tweet contents for enhanced event detection in Twitter. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; Istanbul, Turkey; 2012. pp. 20–24. doi:10.1109/ASONAM.2012.14

[32] Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 2009; 22 (10): 1345–1359. doi:10.1109/TKDE.2009.191

[33] Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing; Doha, Qatar; 2014. pp. 1532–1543. doi:10.3115/v1/D14-1162

[34] Şahinuç F, Toraman C, Koç A. Topic detection based on deep learning language model in Turkish microblogs. In: 29th Signal Processing and Communications Applications Conference (SIU); 2021. pp. 1–4. doi:10.1109/SIU53274.2021.9477781

[35] Sankaranarayanan J, Samet H, Teitler BE, Lieberman MD, Sperling J. TwitterStand: News in tweets. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems; Seattle, Washington; 2009. pp. 42–51. doi:10.1145/1653771.1653781

[36] Soboroff I, Ounis I, Macdonald C, Lin JJ. Overview of the TREC 2012 microblog track. In: Proceedings of the Twenty-First Text REtrieval Conference (TREC 2012); Gaithersburg, Maryland, USA; 2012.

[37] Toraman C. Early prediction of public reactions to news events using microblogs. In: Seventh BCS-IRSG Symposium on Future Directions in Information Access; Barcelona, Spain; 2017. pp. 1–4. doi:10.14236/ewic/FDIA2017.4

[38] Toraman C, Can F, Koçberber S. Developing a text categorization template for Turkish news portals. In: IEEE 2011 International Symposium on Innovations in Intelligent Systems and Applications; Istanbul, Turkey; 2011. pp. 379–383. doi:10.1109/INISTA.2011.5946096

[39] Tsagkias M, Rijke de M, Weerkamp W. Linking online news and social media. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining; Hong Kong, China; 2011. pp. 565–574. doi:10.1145/1935826.1935906

[40] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems; Long Beach, California, USA; 2017. pp. 5998–6008.

[41] Wang J, Tong W, Yu H, Li M, Ma X et al. Mining multi-aspect reflection of news events in Twitter: Discovery, linking and presentation. In: Proceedings of IEEE International Conference on Data Mining; Atlantic City, NJ; 2015. pp. 429–438. doi:10.1109/ICDM.2015.112

[42] Wu Y, Schuster M, Chen Z, Le QV, Norouzi M et al. Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR, 2016, abs/1609.08144.

[43] Yilmaz ZA, Wang S, Yang W, Zhang H, Lin J. Applying BERT to document retrieval with Birch. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing: System Demonstrations; Hong Kong, China; 2019. pp. 19–24. doi:10.18653/v1/D19-3004

[44] Zamani H, Croft WB. Estimating embedding vectors for queries. In: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval; Newark, Delaware, USA; 2016. pp. 123–132. doi:10.1145/2970398.2970403

[45] Zheng G, Callan J. Learning to reweight terms with distributed representations. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval; Santiago, Chile; 2015. pp. 575–584. doi:10.1145/2766462.2767700

[46] Zobel J, Moffat A. Inverted files for text search engines. ACM Computing Surveys 2006; 38 (2): 6–es.