Journal of Universal Computer Science, vol. 28, no. 4 (2022), 345-377
submitted: 29/5/2021, accepted: 26/1/2022, appeared: 28/4/2022 CC BY-ND 4.0

TwitterBulletin: An Intelligent and Real-Time Automated News Categorization Tool for Twitter

Sedef Demirci
(Gazi University, Ankara, Turkey
https://orcid.org/0000-0001-9693-1827, sedefgunduz@gazi.edu.tr)

Seref Sagiroglu
(Gazi University, Ankara, Turkey
https://orcid.org/0000-0003-0805-5818, ss@gazi.edu.tr)

Abstract: Social media platforms have become popular news sources thanks to their immense popularity and high speed of information dissemination. Using these platforms is essential for news organizations and journalists to track and discover news in digital journalism age. However, the abundance of meaningless data and the lack of organization on these platforms make it difficult to reach valuable news for journalists. In this paper, we create the first public dataset containing large number of real-world Turkish news tweets belonging to different news categories, to the best of our knowledge. We propose an artificial intelligence-based two-step approach to assist journalists for accessing the news shared by various sources on social media under the relevant categories like politics (elections, riots, etc.), health (pandemic, covid-19, etc.), etc. via a single platform by reducing the possibility of overlooking needed information. In the first step, we propose a machine learning based novel model for collecting and categorizing news posts on social media. We implement several traditional machine learning and deep learning based algorithms and evaluate their classification performance in terms of accuracy, precision, recall, and F1 score. In the second step, we develop a software tool, named TwitterBulletin, which automatically retrieves Turkish news tweets and groups them under news categories in real time by using the CNN classifier which achieves the best performance in the first step. The results show that the overall accuracy rate of TwitterBulletin is reasonably high and satisfactory despite the challenge of classifying short tweets.

Keywords: News classification, artificial intelligence, deep learning, social media, Twitter, news topic modelling, news dataset
Categories: H.3, I.2, I.7, J.4, L.2, L.1
DOI: 10.3897/jucs.69377

1 Introduction

Social networks are communities that allow people to be aware of each other, share profile information on the Internet, and maintain social relationships. In today's world, social networks are becoming an increasingly popular means of communication. They are not only used for social relations but also for following the latest news and comments about current events [Castillo et al., 2014; Jackoway et al., 2011; Lau, 2014; Vermeer et al., 2020]. In a recent survey, it is found that 51% of the respondents use social media every week to access news [Jiao et al., 2020]. Because news is broken on social media first and can spread faster than on traditional media, social networks are widely used by official news agencies and journalists. Most news organizations expect

journalists to track social media platforms effectively in order to discover and distribute news. However, filtering the posts to obtain only useful and relevant information quickly is one of the most important challenges since different user groups on social media share news in many subject areas [Günther and Quandt, 2016]. In addition, missing requested news information in the continuous and dynamic data flow is among the problems faced by journalists in the age of digital journalism [Hu et al., 2012; Jackoway et al., 2011].

Accordingly, developing artificial intelligence based news categorization systems by using social media data has become a popular research topic. There are various studies in the literature categorizing social media news in English by using traditional machine learning algorithms to find out which algorithm would show the best performance [Dilrukshi et al., 2013; Irfani et al., 2018; Jang et al., 2019; Trieu et al., 2017]. However, none of these systems proposes a fully automatic software for categorizing and presenting social media news in real-time. The news categorization systems developed for Turkish language, on the other hand, do not consider social media as a news source [Acı and Çırak, 2019; Çelik and Koç, 2021; Erdinç and Güran, 2019; Nergiz et al., 2019; Tüfekci et al., 2012; Uslu and Akyol, 2021]. They focus on categorizing long-form newspaper articles on websites of the news agencies, which is considerably different from dealing with short news on social media, in many aspects [Song et al., 2014]. Also, none of them proposes a fully automatic and a real-time system for the end users such as journalists or internet users [Yolcu and Demirci, 2021].

To fill this research gap, in this study, we propose an artificial intelligence based two-step approach, as shown in Figure 1, for categorizing news posts which are shared by the credible users on social media to overcome the aforementioned limitations.

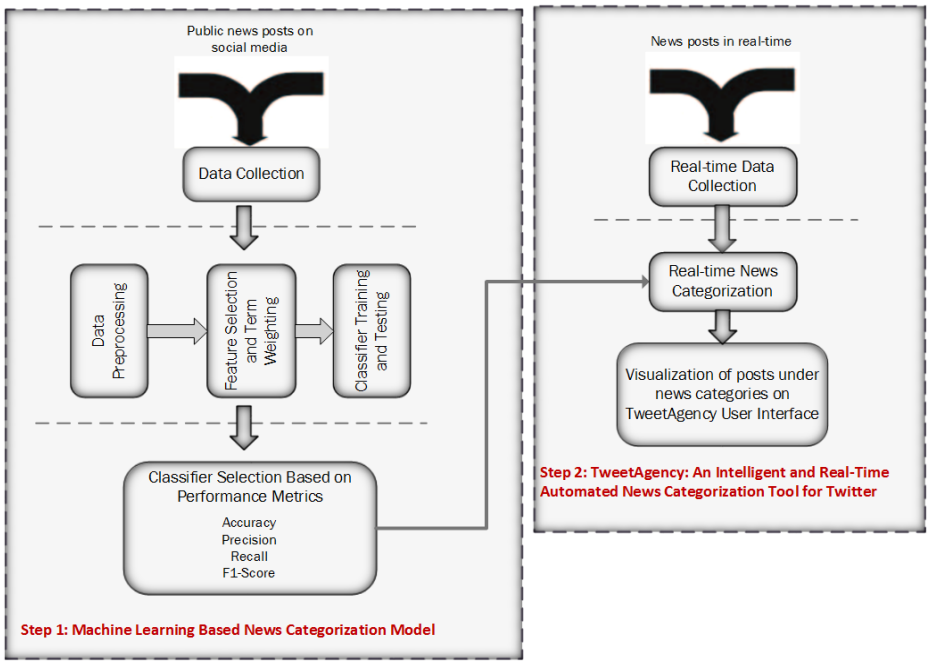

Figure 1: The architecture of the two-step approach for news classification

This study contributes to the literature regarding the following practical and theoretical aspects:

- This is the first attempt, to the best of our knowledge, proposing an intelligent system that collects, classifies and visualizes news tweets in real-time to track and discover the news posts on social media effectively.
- In this regard, we create a real-world dataset of news tweets in Turkish language for the first time in the literature, to the best of our knowledge.
- It is a challenge to determine the best stem offered by Zemberek, a Turkish natural language processing (NLP) library, encountered in Turkish text classification studies. Especially for agglutinative languages like Turkish, root selection is one of the most important factors affecting classification success. Therefore, we improve the performance of Zemberek, a Turkish natural language processing (NLP) library, by combining it with Levenshtein Distance (LD) algorithm.
- For feature selection purpose, we use TF-IDF, N-Gram, and Word2vec vectorization approaches by implementing each of them with different classification algorithms.
- We develop five traditional machine learning based Naïve Bayes (NB), Random Forest (RF), Support Vector Machines (SVM), C4.5, K-Nearest Neighbor (KNN) classifiers and a deep learning based Convolutional Neural Networks (CNN) model in the classification stage. We evaluate the classification performance of these models by combining each of them with different vectorization approaches.
- We evaluate the effect of LD algorithm on performance of each classifier in terms of accuracy, precision, recall, and F1 score.
- We develop a software tool, named TwitterBulletin, which automatically retrieves and determines the category of Turkish news tweets in real time by using CNN classifier which achieves the best performance.

Our system will allow news agencies and journalists to access the news shared by many different sources on social media via a single platform more easily and quickly even if they do not have an account on that platform. Furthermore, the developed system is designed in a flexible manner so that it can be adapted to any social media platform such as Facebook, LinkedIn, etc, and such systems can be developed for different languages by using the proposed approach in this study.

The rest of the paper is organized as follows. Section II provides existing intelligent systems related to online news content. Section III describes our machine learning based novel news categorization model by presenting data collection, preprocessing, term weighting, and classification stages and presents the evaluation results. In Section IV, TwitterBulletin, our real-time news categorization tool for Twitter is presented, and evaluations are made about the real time performance of TwitterBulletin. Finally, in Section V, the work is concluded, suggestions and directions for future work are given.

2 Existing Intelligent Systems Related to Online News Content

There are many studies in literature involving the application of artificial intelligence techniques to collect, process, and categorize news-related contents automatically and alert end users and journalists by means of software tools [De Grove et al., 2020; Diakopoulos, 2020].

In this study, we categorize existing intelligent systems related to online news content under three classes, as shown in Figure 2: (1) newspaper article classification systems (2) newsworthy event detection systems, and (3) fake news detection systems. Newspaper article classification systems deal with categorizing long-form newspaper articles on websites of the news agencies. Newsworthy event detection systems, on the other hand, use social media posts to detect events from news stream.

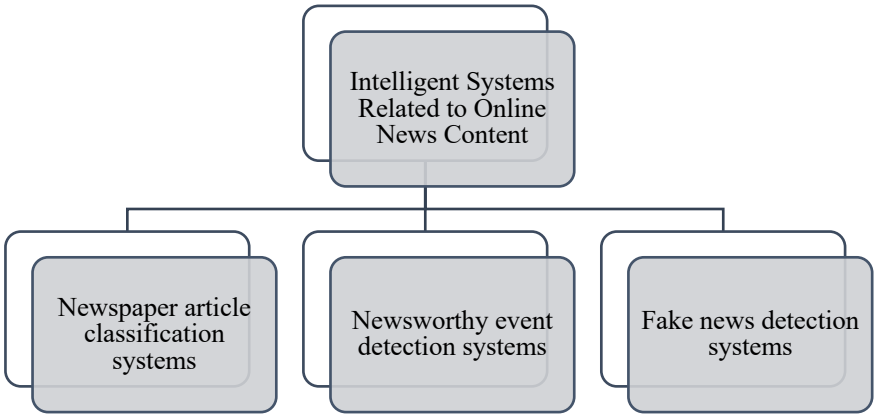

Figure 2: Proposed classification of intelligent systems related to online news content

Fake news detection systems use social media data as well as the newsworthy event detection systems, but they aim to identify fake news to prevent them from spreading.

In the following subsections, we explain the systems for each category in more detail, explain the differences among those systems in terms of research context, and review some examples of state-of-the-art studies, briefly.

2.1 Newspaper article classification systems

It can be said that one of the basic applications of intelligent systems related to news content is classifying newspaper articles on news websites in related news categories to make them more accessible for journalists and the internet users since manual news categorization is a time-consuming, slow and possibly inconsistent task [Toraman et al., 2011]. Classifying newspaper articles is considerably different from dealing with short news on social media. Unlike short news on social media, classical newspaper articles do not contain noise, they are more topic-focused and contain much more words. Therefore, machine learning algorithms relying on word occurrence and

frequency based vectorization approaches produce better accuracy results compared to the approaches classifying short news on social media [Song et al., 2014].

In this regard, Singh et al. [Singh et al., 2020] have used 0.7 million articles from New York Times as a corpus and developed a system that categorizes news articles under 12 different categories such as politics, technology, and sport etc. The highest accuracy level they reach by using Google's BERT model is 87%.

There are also Turkish news classification studies whose data source is daily newspapers. Alparslan et al. [Alparslan et al., 2009] classified Turkish news in three news categories and achieved 92.8% accuracy rate with SVM. In [Toraman et al., 2011], the authors developed a categorization template for Turkish news coming from Bilkent news portal. They concluded that more training data produces better results.

However, none of those newspaper article classification systems considers social media contents, and they do not offer a dynamic system categorizing news articles in real-time.

2.2 Newsworthy event detection systems

According to Diakopoulos, "computational news discovery (CND)" which is a particular area of computational journalism is a recent phenomenon aiming to detect newsworthy events with the use of algorithms. In general, newsworthy event detection systems are based on unsupervised learning, unlike news categorization systems. They generally use unlabelled data and try to cluster events based on the words in social media posts. That means, they differ from short text based news categorization systems in terms of both data structure, and the machine learning approaches [Schwartz et al., 2015].

There are various CND systems developed for detecting newsworthy events, and characterizing and predicting newsworthy events before they happen, in the literature. In this regard, Schwartz et al. [Schwartz et al., 2015] have developed a tool named as CityBeat, which aims to find possible news events and alert journalists in New York City. They have collected social media posts by searching certain keywords and made decision by using machine learning algorithms whether the events are of interest or not. Liu et al. [Liu et al., 2017] have developed a system that identifies news-like events from emerging conversations and presented them to the journalists over a web service. In another study, Nourbakhsh et al. [Nourbakhsh et al., 2017] have presented a system that detects disaster- based news from Twitter in order to help the journalists to analyse the media coverage of disasters. Dilrukshi et al. [Dilrukshi et al., 2013] have tried to detect the most popular news and learn the status of a country via social media posts at any given time.

All of the newsworthy event detection systems presented in this subsection try to detect potential news from the events posted on different social media platforms. However, none of them categorizes news on social media according to their content and propose a real-time classification tool.

2.3 Fake news detection systems

Fake news is the form of news that provides wrong information to the people. Social media platforms are one of the biggest sources of that kind of news since the news on those platforms is not verified compared to the traditional news sources [Keskar et al.,

2020]. Therefore, various computational approaches have proposed in order to detect fake news on social media. Fake news detection systems deal with short, noisy and less topic focused text. However, unlike news categorization systems, meta-data other than text such as user profiles, number of followers, account creation date, etc. are important for those systems. That means, they require different feature representation systems. In addition, fake news detection is generally handled as a binary classification problem, whereas the news categorization on social media is a multi-class classification requiring a completely different classifier design [Aly, 2005].

Özbay and Alataş [Özbay and Alataş, 2020] have utilized pre-annotated datasets and supervised artificial intelligence algorithms to identify them. In another study, Yang et al. [Yang et al., 2019] have investigated the use of unsupervised learning techniques for the situations where annotated datasets are not available. Unlike others, Shu et al. [Shu et al., 2018], have analysed the user profiles and connections among them instead of analysing only the content of the social media posts since detecting fake news only from content would be not satisfactory. However, none of those fake news detection systems consider the category of real news according to their content, and propose a real-time solution when identifying fake news.

From the above analysis, we can conclude that although all of these efforts propose an artificial intelligence based tool for journalism for various purposes, none of them consider automatic monitoring of news on social media for journalistic purposes.

Therefore, to the best of our knowledge, this is the first attempt proposing an intelligent system that collects, classifies and visualizes news tweets in real-time for helping the journalists and news agencies to track and discover the news posts on social media effectively.

3 Proposed News Categorization Model

In this section, the steps of our machine learning based novel news categorization model is explained and the steps of the proposed model is presented in Figure 3.

3.1 Data Collection

Data collection is the first step of creating an intelligent system in order to train and test the system. As mentioned before, we have developed TwitterBulletin for Turkish posts for the time being to show the feasibility of our machine learning based approach. Therefore, we have collected Turkish news tweets via Twitter REST (Representational State Transfer) API, in this step.

We pick 7 news categories to be used in the classification: Technology, Sport, Economy, Politics, Celebrity, Health, and Culture-Art. When deciding on these, categories used by news agencies and other studies in the literature are considered [Dilrukshi et al., 2013; Gupta et al., 2020]. Then, news agencies posting news data from their official Twitter accounts in only one category (such as AgencyA Economy, AgencyA Sport etc.) are selected to avoid manual labelling. Collected tweets from selected accounts are assigned to only one news category. That is, each instance in the dataset is associated with only one target label, making the problem multi-class classification, instead of multi-label classification.

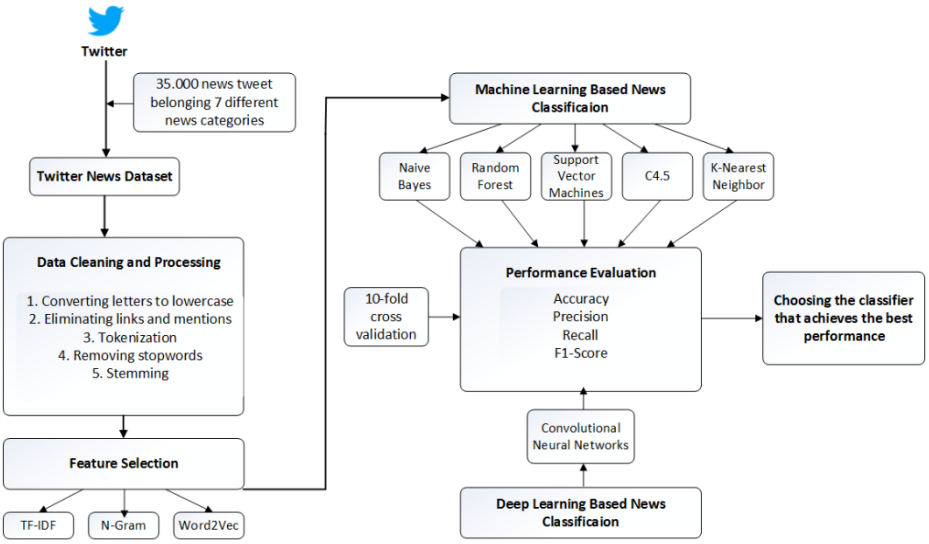

Figure 3: Proposed news categorization model

When collecting tweets from selected accounts, retweets are eliminated since having same tweet would cause memorizing instead of learning.

At the end of this process, total tweet counts for each category are different since tweet frequency varies between categories. This would affect classification performance and accuracy rate of the system negatively, because probability of the presence of a word will be higher for the category from which we collected much more tweet. Then, term weights will be different and learning of the system will be unbalanced for each category. For this reason, total tweet counts for each category are reduced to 5000 by considering the technology category which has the least data. In this process, tweets are sorted by the number of characters and the ones with the maximum number of characters are selected. The purpose of elimination according to character count is training and testing the system with more attributes. Finally, a dataset containing 35000 Turkish news tweets has been created. The dataset is publicly available on http://w3.gazi.edu.tr/~sedefgunduz/train-test-data.txt.

3.2 Data Cleaning and Preprocessing

Algorithm I presents the pseudocode of our data cleaning and preprocessing approach. Data cleaning and preprocessing begins by converting every letter to lowercase. UTF-16 characters are eliminated by means of regular expressions since they do not have any meaning for news tweets. Links and mentions are removed because they are not meaningful words in Turkish [Elsafoury, 2020]. The remaining words are tokenized via whitespace tokenizer. Among the tokenized words, stop words are eliminated since they are heavily used words in a language, do not have a meaning on their own, and have the same meaning for each news class [Uysal and Gunal, 2014]. After obtaining the meaningful words, we need to find their stems since Turkish is an agglutinative language. For this purpose, Zemberek, a Turkish NLP library, is used. Zemberek can

find word stems by using tree implementations constructed with every word in Turkish [Eminağaoğlu and Gökşen, 2020]. If it generates ambiguous outputs for correctly spelled words due to analysing inflections and derivational suffixes in many forms, the first suggestion is accepted by default [Karadağ and Takçı, 2010]. Our contribution is that if Zemberek cannot offer an exact stem for a word because of misspellings, that is, if it offers more than one stem, Levenshtein Distance (LD) is calculated between the original word and the offered stems, and the offer that has the minimum LD to the original word is accepted as the stem.

Algorithm I. Data Cleaning and Preprocessing

tweet ← news tweet retrieved from database
foreach ***tweet*** in database
  convert every letter to lowercase.
  eliminate UTF-16 characters.
  remove links and mentions.
  tokenize each word.
  eliminate stop words.
  find stems of the remaining words with *Zemberek*.
  if *Zemberek* offers an exact stem for a word **then**
    accept the stem and record it to the database
  end if
  else *Zemberek* offers more than one stems for a word **then**
    calculate *Levenshtein Distance (LD)* between the word and each offered stem
    accept the stem that has the minimum *LD* to the original word and record it to the database.
  end if
end for

LD algorithm calculates the distances between letters while measuring the similarity between two words [Levenshtein, 1966]. In order to explain how LD algorithm works with an example, let's assume that we have 2 words in Turkish which are "kitap" (meaning "book" in English) and "hitap" (meaning "declamation" in English). When we spell these words as:

- k-i-t-a-p
- h-i-t-a-p

we see that it is required to change letter "k" with letter "h" in order to equalize these two words. Since we make only one change in this process, Levenstein Distance between these words is calculated as 1.

Finally, obtained words are recorded in the database to be used in further steps.

3.3 Feature Selection and Term Weighting

In this step, a weight value to each word is assigned for each category in order to determine the significance level of each word for each category before developing the classification model [Alsaeedi, 2020]. In text classification, each text-based document should be represented as a vector in vector space model. According to this model, every

tweet corresponds to a document while every word in a tweet corresponds to a term. Documents and terms are represented in Equation (1) and (2) [Alsaeedi, 2020]:

$$D = (d1, d2, d3, \ldots\ldots, dn) \quad (1)$$

$$d1 = (t1, t2, t3, t4, t5, \ldots\ldots, tm) \quad (2)$$

In this study, Term Frequency-Inverse Document Frequency (TF-IDF), N-Gram, and Word2vec vectorization methods are used to represent each tweet in vector space model.

According to TF-IDF method, if a term is seen many times in a few documents in the category to which it belongs, this means that term is significant for that category. TF calculates the frequency of a term in every tweet for each category, while IDF finds the scarcity of a term in all tweets. After calculating the TF and IDF for each word in the dataset, they are multiplied to obtain the TF-IDF value [Alsaeedi, 2020]. The formulas for this method are given in Equation (3), (4) and (5). Here, i corresponds to each term while j is for each tweet.

$$TF - IDFi, j \; = \; TF \; i, j \; x \; IDFi \quad (3)$$

$$TFi, j = \frac{ni, j}{\sum k \; nk, j} \quad (4)$$

$$IDFi = log \frac{|D|}{|\{d : ti \in d\}|} \quad (5)$$

The N-Gram models represent the set of co-occurring words [Brown et al., 1992]. For example, while “social media” is a bigram (for N=2), “social media news” is a trigram (for N=3). In this study, bigram and trigram models are implemented which are mostly preferred N-Gram models in text classification [Karayiğit et al., 2021].

Word2vec is a word embedding system that groups words having close meanings in the vector space. It uses two different learning architectures to establish semantic relationships, which are (i) Continuous Bag of Words (CBoW) and (ii) Skip-Gram. In CBoW architecture, the word in the center of the window is tried to be predicted by looking at the surrounding words. Skip-Gram works similarly to CBoW, but unlike CBoW, it predicts the surrounding words by using the word in the center [Çelik and Koç, 2021]. In this study, CBoW architecture of Word2vec is used to model semantic linguistic features.

3.4 Classification

In classification process, Naïve Bayes, Random Forest, Support Vector Machines, C4.5, and K-Nearest Neighbor algorithms are used, because they are among the mostly preferred algorithms in text classification and accepted as benchmarks [Kabra and Shrawne, 2020; Karayiğit et al., 2021; Kohavi, 1995; Kotzé et al., 2020; Lim et al., 2020; Mukherjee and Sarkar, 2020; Obiedat, 2020; Vaseeharan and Aponso, 2020; Zaman et al., 2020]. Additionally, a Convolutional Neural Networks (CNN) classifier which is one of the mostly used approaches in text categorization is developed in order

to compare the performance of traditional machine learning algorithms with a deep learning based approach. In the following subsection, these algorithms are explained and experimental evaluations are presented.

Naïve Bayes (NB) classifier is easy to design and implement, and produces high success rates and most effective results in terms of time and performance, especially in text classification [Alsaeedi, 2020; Mukherjee and Sarkar, 2020; Vaseeharan and Aponso, 2020]. It is a probability based classifier and assigns a sample into a category with a probability value indicating its confidence by using Bayes theorem [Brindha et al., 2016]. It calculates the probability of assigning a document to all categories by using the formula in Equation (6) [Mandal and Sen, 2014]:

$$p(C|w_1, w_2, w_3, \dots w_n) = log\big(p(C_i)\big) + \sum_{j=1}^{n} log\ (p(w_j|C_i)) \qquad (6)$$

where w represents document features and C represents the categories. NB algorithm makes the assumption that all features of a sample are independent from each other for a class [Chavan et al., 2014].

Random Forest (RF) is a tree based algorithm and it builds a number of random decision trees from the training set which are represented in the form of rules. It is a highly preferred algorithm for text classification since the training process is very fast when compared to other techniques such as neural networks [Kowsari et al., 2019].

The formula behind RF algorithm is presented in Equation (7).

$$n_{ij} = w_l C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \qquad (7)$$

where $n_{isub(j)}$ represents the importance of node j, $left(j)$ is the child node which is split from left part of node j, $right(j)$ is the child node which is split from right part of node j, $w_{sub(j)}$ is the weighted number of samples reaching node j, and $C_{sub(j)}$ represents the impurity value of node j. [Shah et al., 2020].

RF algorithm works in these steps [Kowsari et al., 2019; Sathyadevan and Nair, 2015]:

(1) k random samples are selected from the dataset.
(2) A decision tree is constructed for every sample, and prediction results are obtained from every decision tree.
(3) Then, voting operation is performed for every predicted result.
(4) In the last step, the prediction result which is mostly voted is selected as the final result.

Support Vector Machine (SVM) is one of the mostly used supervised machine learning algorithms in text classification thanks to their robustness against high dimensionality in text data. SVM is based on finding linear separations between various classes [Allahyari et al., 2017; Chavan et al., 2014]. When finding linear separations, SVM creates a hyperplane which separates data points with the maximum margin. Mathematical formula behind this process can be expressed as in Equation (8) [Mandal and Sen, 2014]:

$$y = wx + b \qquad (8)$$

Since SVM algorithm is traditionally used for binary classification, multi category classification problems are decomposed into n binary classification problems while implementing it. One of the most popular approaches is to build one-against-all SVMs for multi-category classification problems, which is used in this study [Kowsari et al., 2019].

C4.5 is a decision tree based statistical classifier which is an extension of the ID3 algorithm [Chavan et al., 2014; Sathyadevan and Nair, 2015]. As different from RF, it builds only one tree for the input data. Branches of the decision tree is determined based on the entropy rules. Entropy is defined as the measure of uncertainty in a system. In the context of C4.5, it refers to how much variance the dataset has. The formula for calculating entropy is given in Equation (9) [Kiliç et al., 2015]:

$$Entropy(C) = -\sum_{i=1}^{n} log_2(p_i) \qquad (9)$$

By using this formula, the entropy for each class i is calculated. Then, gain which is used to quantify the quality of the splits in the tree is calculated by computing the difference between entropy before and after splitting via the formula in Equation (10):

$$Gain(B, C) = Entropy(C) - \sum_{i=1}^{n} \frac{|C_i|}{C} Entropy(C_i) \qquad (10)$$

When determining attribute B for class C, splitting information is used which is formulized in Equation (11):

$$Splitting\ Info(B) = -\sum_{i=1}^{k} \frac{|C_i|}{C} log_2 \frac{|C_i|}{|C|} \qquad (11)$$

In order to determine the splitting criteria when building branches of the decision tree, such as which feature will be the root node etc., gain ratio is considered. The formula for calculating gain ratio is shown in Equation (12):

$$Gain\ Ratio = \frac{Gain\ (B,C)}{Splitting\ Info(B)} \qquad (12)$$

K-Nearest Neighbor (KNN) classifier is easy to implement and it naturally work for multi-category classification problems [Kowsari et al., 2019]. The main idea behind KNN algorithm is to find k nearest neighbors of a test document among all the documents in the train set. It calculates the similarity of the test document to each category, scores the category candidates from the k neighbors and choose the most common category as the class label of the test document. Different distance functions can be used in order to recognize the similarity of the documents, such as Euclidean Distance, Hamming distance, and Minkowski distance etc. The Euclidean Distance which is among the mostly preferred similarity functions in text categorization problems is used in this study. It is calculated by using the following formula, where x and y represents two different data points [Allahyari et al., 2017]:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (13)$$

Convolutional Neural Networks (CNN) is one of the most preferred deep learning methods for text classification. CNN architecture consists of two basic components, which are convolution and pooling layers. In the convolution layer, the filters are shifted over the input data and the results are processed into feature maps. While the input data and filter used during this process are considered as two functions, the shifting process between these functions is called convolution. During this process, representation of the received input is learned. In the pooling layer, the feature maps extracted in the convolution layer are sub-sampled, and in this way, the CNN model learns faster [Acı and Çırak, 2019].

3.5 Evaluation Metrics

The evaluation metrics utilized to compare the performance of the supervised algorithms are accuracy, precision, recall and F1 score, which are determined as follows [Ozbay and Alatas, 2020]:

$$Accuracy = \frac{TP+TN}{\text{TP+FP+TN+FN}} \quad (14)$$

$$Precision = \frac{TP}{\text{TP+FP}} \quad (15)$$

$$Recall = \frac{TP}{\text{TP+FN}} \quad (16)$$

$$F1\ score = \frac{2\ x\ Precision\ x\ Recall}{\text{Precision+Recall}} \quad (17)$$

TP, TN, FP, and FN are four components of the confusion matrix which is taken as a basis of the evaluation of any classifier. They represent the number of true positives, true negatives, false positives, and false negatives respectively.

Classification accuracy refers to the percentage of correctly classified instances by the classifier. It shows how close measurements are made to the true value. However, accuracy alone is not sufficient to measure the performance of a classifier. In order to make more statistical analysis, precision, recall, and F1-score metrics are used. Precision refers to how close measurements are made to each other, and it is calculated as the ratio of correctly classified positive examples divided by the total number of positive examples for a class. Recall, on the other hand, is calculated as the ratio of true positives divided by the total number of true positives and false negatives. Finally, F1-score is the harmonic mean of the precision and recall values [Juba and Le, 2019].

3.6 Experimental Results

In text classification, text data should be separated as training and test sets. In this study, cross validation technique is used to perform this separation. In a study, Kohavi stated that optimum fold number for classification problems is 10 [Kohavi, 1995]. Additionally, it is seen that the most commonly chosen fold number is 10 in the literature [Dilrukshi et al., 2013; Kotzé et al., 2020; Levenshtein, 1966; Obiedat, 2020;]. Therefore, our dataset containing 35000 tweets is divided into 10 folds so that every fold has 3500 tweets. In this way, the NB, RF, SVM, C4.5, KNN, and CNN classifiers

are trained with 9 folds and tested with 1 fold 10 times. In order to obtain the final result, all folds are macro-averaged and testing results are presented in detail, in the remainder of this section.

Different types of vectorization methods which are explained in previous sections are combined with the classification algorithms in order to show how linguistic features affect classification accuracy. In addition, in order to evaluate the effect of LD algorithm which is used in stemming step, we performed our experiments in two stages. In the first stage, we choose the first suggestion of Zemberek when it offers more than one stem, and in the second stage we applied LD algorithm to choose the most appropriate stem from multiple suggestions, instead of using the first one.

The hyperparameters of developed classifiers are presented in Table 1. Table 2 through 7 shows the accuracy results of the classifiers.

Classifier	**Hyperparameters**
NB	alpha=0.1
RF	Number of decision trees=50
SVM	C-SVC method with linear kernel
C4.5	Pruning confidence=0.25
KNN	K=5, euclidean distance
CNN	Epoch size=10, optimization=adam, dropout probability=0.2

Table 1: Hyperparameters of NB, RF, SVM, C4.5, KNN, and CNN classifiers

Algorithms	**Initial Accuracy (%)**	**Improved Accuracy with LD (%)**
NB+TF-IDF	90.41	91.61
NB+bigram+TF-IDF	**93.46**	**94.71**
NB+trigram+TF-IDF	92.73	93.03
NB+CBoW	91.07	91.82

Table 2: Accuracy rates of NB classifier combined with different vectorization methods

Algorithms	**Initial Accuracy (%)**	**Improved Accuracy with LD (%)**
RF+TF-IDF	84.21	85.33
RF+bigram+TF-IDF	**89.46**	**90.33**
RF+trigram+TF-IDF	88.79	89.61
RF+CBoW	89.12	90.04

Table 3: Accuracy rates of RF classifier combined with different vectorization methods

Algorithms	Initial Accuracy (%)	Improved Accuracy with LD (%)
SVM+TF-IDF	**89.39**	**90.82**
SVM+bigram+TF-IDF	89.20	90.69
SVM+trigram+TF-IDF	88.97	89.37
SVM+CBoW	88.70	89.16

Table 4: Accuracy rates of SVM classifier combined with different vectorization methods

Algorithms	Initial Accuracy (%)	Improved Accuracy with LD (%)
C4.5 + TF-IDF	78.65	79.35
C4.5+bigram+TF-IDF	80.99	81.54
C4.5+trigram+TF-IDF	78.90	78.99
C4.5+CBoW	**81.21**	**82.72**

Table 5: Accuracy rates of C4.5 classifier combined with different vectorization methods

Algorithms	Initial Accuracy (%)	Improved Accuracy with LD (%)
KNN + TF-IDF	70.93	71.6
KNN+bigram+TF-IDF	74.67	75.91
KNN+trigram+TF-IDF	73.48	74.59
KNN+CBoW	**76.30**	**76.38**

Table 6: Accuracy rates of KNN classifier combined with different vectorization methods

Algorithms	Initial Accuracy (%)	Improved Accuracy with LD (%)
CNN+TF-IDF	96.00	96.29
CNN+bigram+TF-IDF	96.06	97.06
CNN+trigram+TF-IDF	95.65	96.06
CNN+CBoW	**97.36**	**98.04**

Table 7: Accuracy rates of CNN classifier combined with different vectorization methods

From Table 2-7, it is seen that accuracy rates of all classifiers range from 71% to 98%. CNN algorithm has outperformed other classifiers with the accuracy rate of 98.04,

when it is combined with CBoW and LD algorithms. The second-best accuracy is achieved by the NB+bigram+TF-IDF classifier with the accuracy of 94.71%. NB classifier has achieved its best performance when it is combined with bigram approach instead of CBoW, unlike CNN. The SVM+TF-IDF+LD classifier has performed the third-best accuracy results after CNN and NB algorithms with the accuracy rate of 90.82%. Although bigram representation produces better results for NB algorithm, it does not show the best performance with SVM. However, SVM+bigram+TF-IDF+LD still gives very satisfactory results with 90.69% accuracy rate which is very close to the best result achieved by SVM algorithm. RF algorithm follows SVM with accuracy rate of 90.33% when it is combined with bigram, TF-IDF, and LD approaches. Though the C4.5 and KNN algorithms performed their best when combined with CBoW approach, C4.5+CBoW+LD classifier outperforms KNN+CBoW+LD with the accuracy rate of 82.72%, while this rate is 76.38% for KNN+CBoW+LD.

In addition, from these tables it is clearly shown that LD algorithm improves accuracy rates for all of the classifiers combined with different vectorization algorithms. The rates of increase for all of the classifiers range from 0.08% to 1.51%. While the biggest improvement is seen for C4.5+CBoW classifier, the lowest increase rate is seen for KNN+CBoW. The average increase rates of LD algorithm for NB, RF, SVM, C4.5, KNN, and CNN algorithms that are combined with different vectorization algorithms are 0.875%, 0.9325%, 0,945%, 0.7125%, 0.775%, and 0.595%, respectively. Although they seem as not a big difference, it may vary depending on data set. Our data set does not contain too much misspelled words since they are news. However, LD may be more effective on data sets containing much more tweets from real people.

Additionally, we calculated precision, recall and F1 score for each category in order to statistically evaluate the performance of the developed classifiers. The macro-averaged precision, recall and F1 score results obtained from 10-fold cross validation process for NB, RF, SVM, C4.5, KNN, and CNN classifiers are presented in Table 8, 9, 10, 11, 12, 13, respectively. In the tables, each category name is represented by the initials due to the space constraints (T: technology, S: sport, E: economy, P: politics, C: celebrity, H: health, and C-A: culture-art). Also, performance metrics which are precision, recall, and F1-score are abbreviated as Pr, R, and F1, respectively.

Also, macro-averaged precision, recall and F1-score metrics of all categories are computed in order to be able to combine per-class scores into a single number for each classifier. Since every class has equal number of instances, macro-averaging score is preferred to treat each class equally [Forman, 2004]. As shown in Table 8-19 that LD algorithm has a positive effect on precision, recall, and macro-averaged F1-score metrics for all of the classifiers combined with different vectorization approaches.

The results show that CNN classifier has produced the best performance results among other classifiers in terms of precision, recall and F1-score. CNN has achieved its best performance when it is combined with CBoW and LD approaches. It is also observed that CNN classifier produces the highest F1 score for politics category meaning that it is more precise and robust for politics news. NB algorithm combined with TF-IDF, bigram, and LD approaches produces second-best results with F1-score of 0.9506. It gives the highest precision and recall results for politics and health categories as CNN. The SVM+TF-IDF+LD classifier has the third-best performance results with F1-score of 0.909, but it produces best the F1 score for sport category, unlike CNN and NB algorithms. RF algorithm that achieves its best performance with

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
NB+TF-IDF	Pr	**0.9469**	0.9311	0.9071	0.9244	0.8596	0.8837	0.8981	0.9073
	R	0.9088	0.93	0.896	0.9098	0.8438	**0.9834**	0.8566	0.9041
	F1	0.9274	0.9305	0.9015	0.917	0.8516	**0.9306**	0.8768	0.9051
NB+bigram+TF-IDF	Pr	0.9641	0.9532	0.9368	**0.9938**	0.9053	0.9201	0.933	**0.9438**
	R	0.9394	0.9526	0.9278	0.9384	0.8952	**0.9886**	0.9002	**0.9346**
	F1	0.9516	0.9529	0.9323	**0.9653**	0.9002	0.9531	0.9163	**0.9388**
NB+trigram+TF-IDF	Pr	0.9595	0.9453	0.9328	**0.9937**	0.8889	0.9155	0.9317	0.9382
	R	0.9286	0.9434	0.9134	0.9378	0.8944	**0.988**	0.8892	0.9278
	F1	0.9438	0.9443	0.923	0.9649	0.8916	**0.9504**	0.9099	0.9326
NB+CBoW	Pr	0.949	0.9366	0.9193	**0.9907**	0.8642	0.8934	0.8988	0.9217
	R	0.9264	0.9372	0.9022	0.9184	0.8448	**0.9842**	0.8618	0.9107
	F1	0.9376	0.9369	0.9107	**0.9532**	0.8544	0.9366	0.8799	0.9156

Table 8: The precision, recall and F1-score results of the NB classifier combined with different vectorization approaches

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
RF+TF-IDF	Pr	**0.889**	0.8701	0.8451	0.8623	0.7898	0.8431	0.8313	0.8474
	R	0.8482	0.867	0.8272	0.8594	0.7822	**0.9266**	0.7844	0.8421
	F1	0.8685	0.8686	0.8361	0.8609	0.786	**0.8829**	0.8072	0.8443
RF+bigram+TF-IDF	Pr	**0.9278**	0.9115	0.9007	0.9086	0.8592	0.8906	0.8902	**0.8984**
	R	0.8974	0.9088	0.8838	0.9064	0.8544	**0.9506**	0.861	**0.8946**
	F1	0.9124	0.9102	0.8922	0.9075	0.8568	**0.9196**	0.8754	**0.8963**
RF+trigram+TF-IDF	Pr	**0.9254**	0.8969	0.8936	0.9007	0.8566	0.8861	0.887	0.8923
	R	0.8902	0.9046	0.8716	0.9048	0.8456	**0.9462**	0.8522	0.8879
	F1	0.9074	0.9007	0.8825	0.9027	0.8510	**0.9152**	0.8692	0.8898
RF+CBoW	Pr	**0.9275**	0.9108	0.8958	0.9075	0.8491	0.8878	0.8895	0.8954
	R	0.895	0.9042	0.8824	0.9028	0.8544	**0.9496**	0.8502	0.8912
	F1	0.9109	0.9075	0.8891	0.9052	0.8518	**0.9177**	0.8694	0.8931

Table 9: The precision, recall and F1-score results of the RF classifier combined with different vectorization approaches

TF-IDF, bigram, and LD follows SVM with a fairly close F1-score of 0.905. C4.5 and KNN classifiers produce best performance when they are combined with CBoW and LD approaches, and both of them produces higher F1 score values for technology category. However, C4.5 shows much better performance than KNN in terms of all metrics. It is also observed that classifiers improved with LD approach produce higher F1 scores for sport, technology, and politics categories, in general. This situation is due to the posts belonging to these categories commonly contain distinctive words when

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
SVM+TF-IDF	Pr	**0.9369**	0.9206	0.8971	0.9144	0.8472	0.8759	0.8873	**0.897**
	R	0.9206	0.92	0.8876	0.9038	0.8338	**0.9736**	0.8364	**0.8936**
	F1	0.918	0.9203	0.8923	0.9091	0.8404	**0.9221**	0.8611	**0.8948**
SVM +bigram+TF-IDF	Pr	**0.9353**	0.919	0.8955	0.913	0.8455	0.8744	0.8856	0.8955
	R	0.8968	0.917	0.886	0.9022	0.832	**0.9736**	0.8364	0.892
	F1	0.9157	0.918	0.8907	0.9076	0.8387	**0.9214**	0.8603	0.8932
SVM +trigram+TF-IDF	Pr	**0.9349**	0.9181	0.8924	0.9073	0.8419	0.8735	0.8845	0.8932
	R	0.8938	0.9148	0.8842	0.9008	0.8296	**0.9712**	0.8334	0.8897
	F1	0.9139	0.9165	0.8883	0.9041	0.8357	**0.9198**	0.8582	0.8909
SVM +CBoW	Pr	**0.9317**	0.9152	0.8898	0.9047	0.8395	0.8717	0.8817	0.8906
	R	0.8924	0.9132	0.8818	0.8976	0.8266	**0.9672**	0.8304	0.887
	F1	0.9116	0.9142	0.8858	0.9011	0.833	**0.917**	0.8553	0.8883

Table 10: The precision, recall and F1-score results of the SVM classifier combined with different vectorization approaches

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
C4.5+TF-IDF	Pr	**0.8372**	0.8127	0.7822	0.808	0.7319	0.7895	0.78	0.7916
	R	0.799	0.8078	0.758	0.7902	0.7246	**0.8804**	0.7458	0.7865
	F1	0.8176	0.8102	0.7699	0.799	0.7282	**0.8325**	0.7625	0.7886
C4.5 +bigram+TF-IDF	Pr	**0.8598**	0.8352	0.8051	0.8312	0.7593	0.8081	0.8024	0.8145
	R	0.827	0.8332	0.7898	0.8012	0.7426	**0.9002**	0.775	0.8099
	F1	0.8431	0.8342	0.7974	0.816	0.7509	**0.8517**	0.7885	0.8117
C4.5 +trigram+TF-IDF	Pr	**0.8395**	0.8153	0.7837	0.8104	0.734	0.7913	0.7843	0.7941
	R	0.8014	0.8104	0.7608	0.7924	0.7274	**0.8832**	0.7474	0.789
	F1	0.82	0.8128	0.7721	0.8013	0.7307	**0.8347**	0.7654	0.791
C4.5 +CBoW	Pr	**0.8604**	0.8358	0.8085	0.8339	0.7621	0.811	0.8044	**0.8166**
	R	0.8416	0.833	0.7904	0.8082	0.742	**0.896**	0.7738	**0.8121**
	F1	0.8509	0.8344	0.7994	0.8208	0.7519	**0.8514**	0.7888	**0.8139**

Table 11: The precision, recall and F1-score results of the C4.5 classifier combined with different vectorization approaches

Classifier/Category		T	S	E	P	C	H	C-A	Macro-average
KNN+TF-IDF	Pr	**0.7592**	0.7313	0.6924	0.7413	0.6654	0.7149	0.7057	0.7157
	R	0.727	0.7268	0.6864	0.7202	0.6426	**0.7884**	0.6738	0.7093
	F1	0.7427	0.7291	0.6894	0.7306	0.6535	**0.7499**	0.6894	0.7121
KNN+bigram+TF-IDF	Pr	**0.7971**	0.7685	0.7344	0.7766	0.7029	0.7508	0.7378	0.7526
	R	**0.8228**	0.7826	0.7244	0.7252	0.6834	0.8028	0.686	0.7467
	F1	**0.8098**	0.7755	0.7294	0.75	0.693	0.776	0.711	0.7492
KNN+trigram+TF-IDF	Pr	**0.7894**	0.7452	0.7204	0.7676	0.6933	0.7425	0.7263	0.7407
	R	**0.8166**	0.7758	0.7044	0.72	0.6688	0.7876	0.6702	0.7348
	F1	**0.8028**	0.7602	0.7123	0.743	0.6809	0.7644	0.6971	0.7372
KNN+CBoW	Pr	**0.8089**	0.7869	0.7541	0.7916	0.7202	0.7626	0.7542	**0.7684**
	R	**0.8438**	0.8044	0.7342	0.7514	0.6878	0.8082	0.7112	**0.763**
	F1	**0.826**	0.7956	0.744	0.771	0.7036	0.7847	0.7321	**0.7653**

Table 12: The precision, recall and F1-score results of the KNN classifier combined with different vectorization approaches

Classifier/Category		T	S	E	P	C	H	C-A	Macro-average
CNN+TF-IDF	Pr	0.9833	0.9775	0.9647	**0.997**	0.9376	0.9465	0.9489	0.9651
	R	0.965	0.9752	0.9558	0.9654	0.9226	**0.9976**	0.9388	0.9601
	F1	0.9741	0.9764	0.9602	**0.9809**	0.93	0.9714	0.9438	0.9624
CNN+bigram+TF-IDF	Pr	0.9837	0.9778	0.9657	**0.997**	0.938	0.9472	0.9494	0.9655
	R	0.9658	0.976	0.957	0.9658	0.9226	**0.9982**	0.9388	0.9606
	F1	0.9747	0.9769	0.9613	**0.9811**	0.9302	0.9721	0.9441	0.9629
CNN+trigram+TF-IDF	Pr	0.9824	0.9757	0.9614	**0.9968**	0.9316	0.9417	0.944	0.9619
	R	0.9618	0.973	0.952	0.9624	0.9154	**0.9976**	0.933	0.9565
	F1	0.972	0.9744	0.9567	**0.9793**	0.9234	0.9688	0.9384	0.959
CNN+CBoW	Pr	0.9917	0.9887	0.9787	**0.9986**	0.955	0.9648	0.9594	**0.9767**
	R	0.9784	0.984	0.9748	0.981	0.9432	**0.999**	0.955	**0.9736**
	F1	0.985	0.9864	0.9768	**0.9897**	0.9491	0.9816	0.9572	**0.9751**

Table 13: The precision, recall and F1-score results of the CNN classifier combined with different vectorization approaches

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
NB+TF-IDF+LD	Pr	**0.9514**	0.9376	0.9162	0.9321	0.8874	0.8991	0.9051	0.9184
	R	0.9312	0.9644	0.892	0.9012	0.8574	**0.9834**	0.8832	0.9161
	F1	0.9412	**0.9508**	0.9039	0.9164	0.8721	0.9393	0.894	0.9168
NB+bigram+TF-IDF+LD	Pr	0.9744	0.9673	0.9482	**0.996**	0.921	0.935	0.9399	**0.9545**
	R	0.9514	0.9692	0.938	0.9478	0.9092	**0.9952**	0.9192	**0.9471**
	F1	0.9628	0.9682	0.9431	**0.9713**	0.9151	0.9642	0.9294	**0.9506**
NB+trigram+TF-IDF+LD	Pr	0.964	0.9456	0.939	**0.9937**	0.8886	0.9205	0.9319	0.9405
	R	0.932	0.9482	0.9168	0.9424	0.8934	**0.989**	0.892	0.9305
	F1	0.9477	0.9469	0.9277	**0.9674**	0.891	0.9535	0.9115	0.9351
NB+CBoW+LD	Pr	0.9511	0.9382	0.9216	**0.9908**	0.8818	0.9072	0.902	0.9275
	R	0.9292	0.939	0.91	0.9256	0.8548	**0.9852**	0.8834	0.9182
	F1	0.94	0.9386	0.9158	**0.9571**	0.8681	0.9445	0.8926	0.9224

Table 14: The precision, recall and F1-score results of the NB classifiers improved with LD approach

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
RF+TF-IDF+LD	Pr	**0.8989**	0.8827	0.8571	0.8681	0.8032	0.8569	0.8407	0.8582
	R	0.8586	0.8772	0.8382	0.8702	0.7932	**0.9386**	0.797	0.8533
	F1	0.8783	0.8799	0.8475	0.8692	0.7981	**0.8959**	0.8183	0.8553
RF +bigram+TF-IDF+LD	Pr	**0.9353**	0.9173	0.9097	0.9144	0.8779	0.8998	0.8943	**0.9069**
	R	0.9168	0.927	0.888	0.908	0.8574	**0.9516**	0.8744	**0.9033**
	F1	**0.926**	0.9221	0.8987	0.9112	0.8676	0.925	0.8842	**0.905**
RF +trigram+TF-IDF+LD	Pr	**0.9289**	0.9125	0.9012	0.9125	0.8602	0.8927	0.8915	0.8999
	R	0.8984	0.9098	0.888	0.9072	0.8568	**0.952**	0.861	0.8962
	F1	0.9134	0.9112	0.8945	0.9098	0.8585	**0.9214**	0.876	0.8978
RF +CBoW+LD	Pr	**0.9322**	0.9145	0.9036	0.9144	0.8631	0.8977	0.9	0.9037
	R	0.9054	0.9138	0.8942	0.91	0.8628	**0.9508**	0.8658	0.9004
	F1	0.9186	0.9142	0.8989	0.9122	0.863	**0.9235**	0.8826	0.9018

Table 15: The precision, recall and F1-score results of the RF classifiers improved with LD approach

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
SVM+TF-IDF+LD	Pre	**0.9434**	0.9298	0.908	0.9232	0.8789	0.8932	0.8976	**0.9106**
	R	0.9234	0.9568	0.8838	0.894	0.8494	**0.9748**	0.875	**0.9082**
	F1	0.9333	**0.9431**	0.8957	0.9084	0.8639	0.9322	0.8862	**0.909**
SVM +bigram+TF-IDF+LD	Pr	**0.9423**	0.9284	0.9065	0.9221	0.8775	0.8921	0.8966	0.9094
	R	0.9218	0.955	0.882	0.8924	0.8494	**0.974**	0.874	0.9069
	F1	0.932	**0.9415**	0.8941	0.907	0.8632	0.9313	0.8852	0.9077
SVM +trigram+TF-IDF+LD	Pr	**0.9367**	0.9204	0.8974	0.9149	0.8473	0.8754	0.8881	0.8972
	R	0.8994	0.9208	0.887	0.905	0.8346	**0.9722**	0.8368	0.8937
	F1	0.9177	0.9206	0.8922	0.9099	0.8409	**0.9213**	0.8617	0.8949
SVM +CBoW+LD	Pr	**0.934**	0.9184	0.8946	0.9125	0.8451	0.8735	0.8871	0.895
	R	0.8974	0.9188	0.8846	0.9036	0.8324	**0.9698**	0.8344	0.8916
	F1	0.9153	0.9186	0.8896	0.908	0.8387	**0.9192**	0.8599	0.8928

Table 16: The precision, recall and F1-score results of the SVM classifiers improved with LD approach

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
C4.5+TF-IDF+LD	Pr	**0.8445**	0.8197	0.7891	0.8148	0.7387	0.7954	0.7878	0.7986
	R	0.808	0.8148	0.765	0.7952	0.7316	**0.8894**	0.7508	0.7935
	F1	0.8258	0.8173	0.7769	0.8049	0.7351	**0.8398**	0.7689	0.7955
C4.5 +bigram+TF-IDF+LD	Pr	**0.8659**	0.8399	0.8113	0.8368	0.7637	0.8134	0.807	0.8197
	R	0.8432	0.8354	0.7926	0.8048	0.746	**0.9038**	0.7818	0.8154
	F1	0.8544	0.8377	0.8018	0.8205	0.7548	**0.8562**	0.7942	0.8171
C4.5 +trigram+TF-IDF+LD	Pr	**0.8398**	0.8162	0.7848	0.8115	0.7349	0.7924	0.7852	0.795
	R	0.8022	0.8116	0.7616	0.7936	0.7284	**0.8842**	0.748	0.7899
	F1	0.8206	0.8139	0.773	0.8024	0.7316	**0.8358**	0.7661	0.7919
C4.5 +CBoW+LD	Pr	**0.875**	0.8494	0.8262	0.85	0.7744	0.8261	0. 8195	**0.8315**
	R	0.8554	0.8482	0.8108	0.826	0.7648	**0.9034**	0.7816	**0.8272**
	F1	**0.8651**	0.8488	0.8184	0.8378	0.7696	0.863	0.8001	**0.829**

Table 17: The precision, recall and F1-score results of the C4.5 classifiers improved with LD approach

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
KNN+TF-IDF+LD	Pr	**0.7655**	0.7377	0.7002	0.7472	0.6722	0.7201	0.7127	0.7222
	R	0.734	0.7318	0.6954	0.7252	0.7252	**0.7934**	0.6828	0.716
	F1	0.7494	0.7347	0.6978	0.736	0.736	**0.755**	0.6974	0.7187
KNN+bigram+TF-IDF+LD	Pr	**0.8039**	0.7812	0.7501	0.7883	0.7174	0.7608	0.7496	0.7645
	R	**0.8388**	0.799	0.7318	0.7464	0.6834	0.8058	0.7088	0.7591
	F1	**0.821**	0.79	0.7408	0.7668	0.7	0.7826	0.7286	0.7614
KNN+trigram+TF-IDF+LD	Pr	**0.7959**	0.7659	0.7344	0.7762	0.7034	0.7515	0.7359	0.7519
	R	**0.8228**	0.7798	0.7232	0.7254	0.6824	0.8032	0.6844	0.7459
	F1	**0.8091**	0.7728	0.7287	0.7499	0.6927	0.7765	0.7092	0.7484
KNN+CBoW+LD	Pr	**0.8093**	0.7875	0.7549	0.7926	0.7214	0.7638	0.755	**0.7692**
	R	**0.8438**	0.8036	0.7342	0.7538	0.6898	0.8112	0.712	**0.7641**
	F1	**0.8262**	0.7955	0.7444	0.7727	0.7052	0.7868	0.7329	**0.7662**

Table 18: The precision, recall and F1-score results of the KNN classifiers improved with LD approach

Classifier/Category		T	S	E	P	C	H	C-A	**Macro-average**
CNN+TF-IDF+LD	Pr	0.9879	0.9813	0.9669	**0.9976**	0.9394	0.9503	0.9507	0.9677
	R	0.9672	0.9778	0.9586	0.968	0.9262	**0.9976**	0.9446	0.9629
	F1	0.9775	0.9796	0.9627	**0.9826**	0.9327	0.9734	0.9476	0.9652
CNN+bigram+TF-IDF+LD	Pr	0.9903	0.9873	0.9765	**0.9984**	0.9495	0.9617	0.9546	0.974
	R	0.9758	0.9824	0.9722	0.9784	0.9372	**0.9982**	0.95	0.9706
	F1	0.983	0.9849	0.9743	**0.9883**	0.9433	0.9796	0.9523	0.9722
CNN+trigram+TF-IDF+LD	Pr	0.9841	0.9779	0.9649	**0.997**	0.9384	0.9469	0.9494	0.9655
	R	0.9652	0.9756	0.956	0.9664	0.9236	**0.9984**	0.9388	0.9606
	F1	0.9746	0.9768	0.9604	**0.9815**	0.931	0.972	0.9441	0.9629
CNN+CBoW+LD	Pr	0.9933	0.9908	0.9859	**0.9988**	0.9721	0.9739	0.9675	**0.9832**
	R	0.9826	0.9868	0.9792	0.988	0.9571	**0.999**	0.9718	**0.9806**
	F1	0.9879	0.9888	0.9825	**0.9934**	0.9646	0.9863	0.9697	**0.9819**

Table 19: The precision, recall and F1-score results of the CNN classifiers improved with LD approach

compared to other categories. On the other hand, celebrity and culture-art categories have lower F1 scores for nearly all classifiers since the news posted on these categories are closely related to each other. In addition, the results obtained in this study are compared with the existing literature on news classification in Table 20. In the table, compared performance metrics which are accuracy, precision, recall, and F1-score are abbreviated as A, Pr, R, and F1, respectively.

Reference	Data	Language	Feature	Algorithm	A	Pr	R	F1
[Acı and Çırak, 2019]	News articles	Turkish	Word2vec	CNN	93.3%	-	-	0.95
[Çelik and Koç, 2021]	News articles	Turkish	TF-IDF, Word2vec, Fasttext	SVM, NB, LR, RF, ANN	90-95%	-	-	-
[Dilrukshi et al., 2013]	Tweets	English	TF	SVM	90-99%	0.81-1	0.3-0.89	0.53-0.88
[Dilrukshi and Zoysa 2013]	Tweets	English	TF	NB, SVM	-	-	-	0.38-0.97
[Erdinç and Güran, 2019]	News articles	Turkish	Fasttext, Word2vec, Doc2vec	SVM, NB, LR, DT, ANN	75-78%	-	-	-
[Irfani et al., 2018]	Tweets	English	WordNet	NB	66-72%	-	-	-
[Jang et al., 2019]	News articles and tweets	English	Word2vec	CNN	84-93%	-	-	0.83-0.93
[Nergiz et al., 2019]	News articles	Turkish	Fasttext, Word2vec, Doc2vec	RNN, LSTM	93-96%	-	-	-
[Trieu et al., 2017]	Tweets	English	Doc2vec	KNN	78-94%	-	-	-
[Tüfekci et al., 2012]	News articles	Turkish	TF	NB, SVM, C4.5, RF	73-92%	-	-	-
[Uslu and Akyol, 2021]	News articles	Turkish	-	NB, SVM, RF	-	0.87-0.91	0.87-0.91	0.87-0.91
This work	**Tweets**	**Turkish**	**TF-IDF, N-Gram, Word2vec**	**NB, SVM, RF, C4.5, KNN, and CNN**	**76-98%**	**0.76-98**	**0.76-98**	**0.76-98**

Table 20: Comparison of this work with the existing literature on news classification

As we have mentioned before, this is the first attempt proposing a system classifying short Turkish news tweets in real-time. Therefore, for comparison purposes, existing studies in Table 20 can be evaluated under two categories: (i) studies

classifying long-form news articles in Turkish, and (i) studies classifying short news tweets in English. The accuracy rates obtained in this study are reasonably high and acceptable, when compared to studies classifying Turkish long newspaper articles whose accuracy results range from 73% to 95%. Despite the challenge of classifying short text, developed CNN classifier with the accuracy rate of 98% has produced better results than previous studies using longer newspaper articles.

On the other hand, the accuracy rates of the studies classifying short English news tweets range from 66-99%. Since Turkish is an agglutinative language, preprocessing steps like stemming is more difficult when compared to English language [Karayiğit et al., 2021]. Therefore, despite the challenge of working with Turkish language, the accuracy results obtained in this study are very high and acceptable when compared with the studies using English news posts.

Finally, since we have achieved best performance results with CNN+CBoW+LD classifier, we have developed TwitterBulletin by using this approach. In the following section, we present TwitterBulletin and evaluations about the real time performance of this real-time news classification tool.

4 TwitterBulletin: Machine Learning Based Automated and Real-Time News Categorization Tool

4.1 Real-Time Data Collection and Categorization

Public Twitter accounts that share newsworthy posts such as news agencies, journalists, bureaucrats, government agencies, important private institutions etc. are included in the system and their tweets are collected in real time. In order to perform this work, Twitter Streaming API is used. It is dedicated to developers by Twitter for accessing and recording public tweets in a continuous way [Twitter Developers, 2021]. With Streaming API; id number, name and profile photo of the account, content and time of each tweet (as day-month-year, hour-minute-second), contained links (if exists) are recorded in our database in real-time. The purpose of storing this information is to classify the tweets in real-time and then present them to the journalists over TwitterBulletin with all the metadata of that tweet. Each tweet collected in real-time is assigned to an appropriate news category smartly by the CNN based news classification approach explained above. Then, tweets are recorded to the database with their categories after being classified.

4.2 Visualization on TwitterBulletin

In Figure 4, the main page of TwitterBulletin which runs on the above algorithms is displayed. Main page shows the most recently posted newsworthy tweets shared by the listened Twitter accounts in real-time. That is, when a tweet is posted by an account, it appears in the main page of TwitterBulletin. Also, news category of each tweet calculated by our machine learning based approach is shown next to the tweet in the main page.

Figure 4: Main page of TwitterBulletin

In addition, users can search and track the newsworthy tweets by category. For example, Figure 5 shows the tweets on sport category when user clicks to sport button in order to show the news tweets under this category.

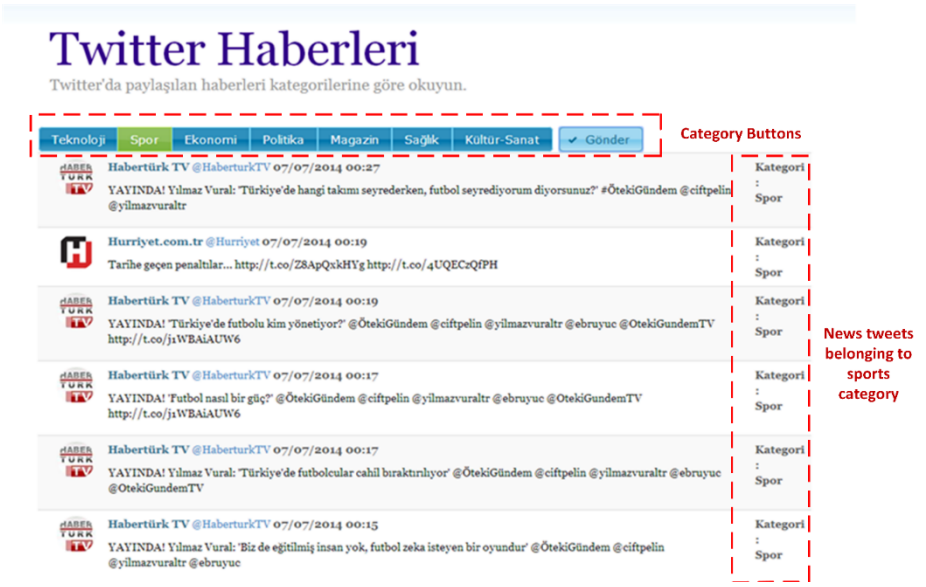

Figure 5: Tweets under sport category

Additionally, category buttons are designed to be multi-selectable so that users can access news tweets from many categories at once. As shown in Figure 6, when a user

selects health and culture-art categories, the tweets under these categories are shown to the user in one page.

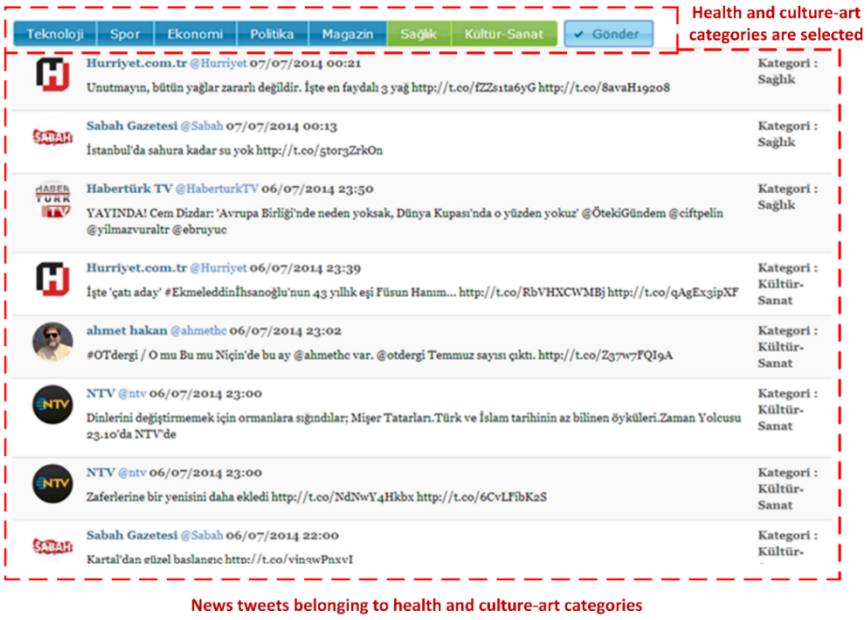

Figure 6: Tweets under health and culture-art categories

4.3 Evaluation of TwitterBulletin in Real-Time

In order to evaluate the performance of TwitterBulletin, we measure the percentage of correctly classified tweets among those collected in real-time. For this purpose, Twitter news accounts, especially those posting news tweets in only one category which are used in training phase are listened for one week and it is checked whether TwitterBulletin could be able to assign these tweets to their actual category in real-time. Table 21 shows the confusion matrix obtained from the real time performance test of TwitterBulletin. In Table 22, which is created based on Table 21, error rates per category are shown. Also, some examples of misclassified tweets are shown in Table 23. As it can be seen from Table 22, correctly classified tweet count for technology category is 205, while the incorrectly classified tweet count is only 4, meaning that misclassification rate for this category is 1.91%. In the same manner, misclassification rates for sport, economy, politics, celebrity, health, and culture-art categories are 0.27%, 2.29%, 3.61%, 3.58%, 1.47%, and 8.16% respectively. Technology sport, and health news is mostly confused with celebrity news with the error rates of 0.48%, 0.27%, and 0.33%, respectively. Celebrity news, on the other hand, are mostly confused with culture-art with 0.75% error rate. This shows that health news contains more common words with celebrity news compared to other categories, most likely due to covid-19 pandemic. Also, economy news is mostly confused with politics news with

		Actual Class						
Predicted Class	**Confusion Matrix**	Technology	Sport	Economy	Politics	Celebrity	Health	Culture-Art
	Technology	205	0	2	1	4	2	0
	Sport	1	373	0	2	5	2	0
	Economy	2	0	298	4	1	2	0
	Politics	0	0	4	240	1	1	0
	Celebrity	1	1	0	0	512	2	4
	Health	0	0	1	2	4	602	0
	Culture-Art	0	0	0	0	4	0	45

Table 21: Confusion matrix obtained from real-time performance test of TwitterBulletin

1.13% misclassification rate, which is an expected situation due to Turkey's recent economy-based politics agenda.

Category	**Correctly classified tweet count**	**Misclassified tweet count**	**Error rate (%)**
Technology	205	4	1.91
Sport	373	1	0.27
Economy	298	7	2.29
Politics	240	9	3.61
Celebrity	512	19	3.58
Health	602	9	1.47
Culture-Art	45	4	8.16

Table 22: Error rates of TwitterBulletin per category

In addition, according to this confusion matrix, the accuracy of TwitterBulletin in real-time is calculated as 97.72% which is only 0.32% lower than the accuracy rate of CNN+CBoW+LD classifier in the training phase. This difference in accuracy depends

on the words in posts used for real-time evaluation purposes and it may stem from the fact that news in Turkey naturally changed from the training time to the time of real-time evaluation. This suggests that the accuracy rate of TwitterBulletin can vary depending on the words included in incoming tweets because the system may face some words that it had not learned in the training phase.

Tweets	**Actual category**	**Predicted category**
"Sosyal medyayı dinlemeyin" DSÖ'den aşı karşıtlarına ilişkin açıklama	Health	Technology
Fatih Terim'den dikkat çeken açıklama: Teklif gelirse hemen ayrılırım	Sport	Celebrity
Aşı olmadığı için konseri iptal edilen Yıldız Tilbe, aşısını oldu	Celebrity	Health
Son noktayı koydu "Yeni James Bond bir kadın mı olacak?"	Culture-Art	Celebrity
'Para politikası yön değiştiriyor'	Economy	Politics

Table 23: Tweet examples that are misclassified by TwitterBulletin

Table 24 presents the precision, recall, and F1 score for each category obtained in the real-time performance test of TwitterBulletin.

Category	**Precision**	**Recall**	**F1-Score**
Technology	0.9309	0.9579	0.9693
Sport	0.9973	0.9739	0.9855
Economy	0.977	0.9707	0.9739
Politics	0.9639	0.9756	0.9697
Celebrity	0.9642	0.9846	0.9743
Health	0.9853	0.9851	0.9869
Culture-Art	0.9184	0.9184	0.9184
Macro-average	**0.9696**	**0.9671**	**0.9683**

Table 24: Precision, recall, and F1-Score of TwitterBulletin in real-time performance test

When these results are compared to the findings in training phase, it is seen that F1 score which is the harmonic mean of the precision and recall values is lower for technology, sport, economy, politics, and culture-art categories; while it is higher for celebrity and health categories. These also suggests the classification performance of TwitterBulletin may change from time to time. In this regard, it is concluded that dynamic training of the system via user feedbacks about incorrectly classified posts may improve the success of the system.

5 Conclusions and Future Directions

In this study, we present an artificial intelligence based two-step approach to assist journalists and news agencies for following the news shared by various sources on social media. In the first step, we propose an artificial intelligence based novel model for collecting, preprocessing and classifying news posts. To this end, we create a real-world dataset of news tweets in Turkish language for the first time in the literature, to the best of our knowledge. In data cleaning and preprocessing step, we improve the performance of Zemberek, a Turkish natural language processing (NLP) library, by combining it with Levenshtein Distance (LD) algorithm. For feature selection purpose, we use TF-IDF, N-Gram, and Word2vec vectorization approaches by implementing each of them with different classification algorithms. We develop five traditional machine learning based classifiers and a deep learning based Convolutional Neural Networks (CNN) model in the classification stage.

In the second step, we develop a software tool, named TwitterBulletin, that determines the category of Turkish news tweets and visualizes them on the user interface in real time. It allows journalists and news agencies to follow the news posts shared by many different sources on Twitter via a single platform and access relevant information effectively and quickly in the age of digital journalism.

The key strengths and contributions of this study filling the research gaps in this area can be summarized in threefold: (i) a real-world dataset of news tweets in Turkish language, (ii) an intelligent system model for collecting, classifying, and visualizing news tweets, and (iii) a software tool, TwitterBulletin, retrieving and categorizing Turkish news tweets automatically and in real time. Our analysis shows that the overall accuracy rate of TwitterBulletin is reasonably high and satisfactory when compared with the studies classifying newspaper articles although Twitter posts contain much less words because of the character limit.

The limitations of our study, on the other hand, are twofold. Firstly, TwitterBulletin supports only Turkish for the time being. However, proposed machine learning based approach can easily be adapted to different languages and also different social media platforms like Facebook. Therefore, adapting the proposed approach to other languages can be the subject of new studies. Secondly, our system does not have a feedback mechanism to receive user suggestions for incorrectly classified posts, and dynamically training the system in line with these feedbacks. In this regard, adding a feedback mechanism to increase the accuracy is among the future work. Adding filtering options to TwitterBulletin by time and importance level, and a search engine for news posts by specified keywords are also among the planned future work. Additionally, conducting a study with longer social media posts and comparing the results with the current system are possible extensions.

References

[Acı and Çırak, 2019] Acı, Ç., Çırak, A.: "Türkçe haber metinlerinin konvolüsyonel sinir ağları ve Word2Vec kullanılarak sınıflandırılması"; Bilişim Teknolojileri Dergisi, 12(3), (2019), 219-228.

[Allahyari et al., 2017] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: "A brief survey of text mining: Classification, clustering and extraction techniques"; (2017), arXiv preprint arXiv:1707.02919.

[Alparslan et al., 2009] Alparslan, E., Bahsi, B., Karahoca, A.: "Classification of Turkish News Documents Using Support Vector Machines"; International IEEE Symposium on Innovations in Intelligent Systems and Applications, Trabzon, Turkey, (2009), 466-471

[Alsaeedi, 2020] Alsaeedi, A.: "A survey of term weighting schemes for text Classification"; International Journal of Data Mining, Modelling and Management, 12(2), (2020), 237-254

[Aly, 2005] Aly, M.: "Survey on multiclass classification methods"; Neural Networks, 19, (2005), 1-9.

[Brindha et al., 2016] Brindha, S., Prabha, K., Sukumaran, S.: "A survey on classification techniques for text mining"; 3rd IEEE International Conference on Advanced Computing and Communication Systems, Coimbatore, India, (2016), 1-5

[Brown et al., 1992] Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., Mercer, R. L.: "Class-based n-gram models of natural language"; Computational linguistics, 18(4), (1992), 467-480.

[Castillo et al., 2014] Castillo, C., El-Haddad, M., Pfeffer, J.: "Characterizing the life cycle of online news stories using social media reactions"; International ACM Conference on Computer Supported Cooperative Work & Social Computing, Baltimore, MD, USA, (2014), 221-223.

[Cataltepe et al., 2007] Cataltepe, Z., Turan, Y., Kesgin, F.: "Turkish document classification using shorter roots"; International IEEE Conference on Signal Processing and Communications Applications, Eskişehir, Turkey, (2007), 1-4

[Çelik and Koç, 2021] Çelik, Ö., Koç, B. C.: "TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması"; Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 23(67), (2021), 121-127.

[Chavan et al., 2014] Chavan, G.S., Manjare, S., Hegde, P., Sankhe, A.: "A survey of various machine learning techniques for text Classification"; International Journal of Engineering Trends and Technology, 15(6), (2014), 288-292

[De Grove et al., 2020] De Grove, F., Boghe, K., De Marez, L.: "(What) Can Journalism Studies Learn from Supervised Machine Learning?"; Journalism Studies, 21(7), (2020), 912-927. https://doi.org/10.1080/1461670X.2020.1743737

[Diakopoulos, 2020] Diakopoulos, N.: "Computational News Discovery: Towards Design Considerations for Editorial Orientation Algorithms in Journalism"; Digital Journalism, (2020), 1-23. ttps://doi.org/10.1080/21670811.2020.1736946

[Dilrukshi et al., 2013] Dilrukshi, I., Zoysa, K.D., Caldera, A.: "Twitter news classification using SVM": International IEEE Conference on Computer Science & Education, Colombo, Sri Lanka, (2013), 287-291.

[Dilrukshi and De Zoysa, 2013] Dilrukshi, I., De Zoysa, K.: "Twitter news classification: Theoretical and practical comparison of SVM against Naive Bayes algorithms"; IEEE International Conference on Advances in ICT for Emerging Regions,Colombo, Sri Lanka, (2013), 278.

[Elsafoury, 2020] Elsafoury, F.: "Teargas, Water Cannons and Twitter: A case study on detecting protest repression events in Turkey 2013"; International Workshop on Narrative Extraction from Texts, Lisbon, Portugal, (2020), 5-13

[Eminağaoğlu and Gökşen, 2020] Eminağaoğlu, M., Gökşen, Y.: "A New Similarity Measure for Document Classification and Text Mining"; KnE Social Sciences, (2020), 353-366. https://doi.org/10.18502/kss.v4i1.5999

[Erdinç and Güran, 2019] Erdinç, H. Y., Güran, A.: "Semi-supervised turkish text categorization with word2vec, doc2vec and fasttext algorithms"; IEEE 27th Signal Processing and Communications Applications Conference, Sivas, Turkey, (2019), 1-4.

[Fletcher et al., 2020] Fletcher, R., Schifferes, S., Thurman, N.: "Building the 'Truthmeter': Training algorithms to help journalists assess the credibility of social media sources"; Convergence, 26(1), (2020), 19-34. https://doi.org/10.1177/1354856517714955

[Forman, 2004] Forman, G.: "A pitfall and solution in multi-class feature selection for text classification"; Twenty-first international conference on Machine learning, Banff Alberta, Canada (2004), 38.

[Gupta et al., 2020] Gupta, D., Sharma, A., Kumar, M.: "TweetsDaily: Categorised News from Twitter"; International Conference on Advances in Computing and Intelligent Systems, Singapore, (2020), 55-67

[Günther and Quandt, 2016] Günther, E., Quandt, T.: "Word counts and topic models: Automated text analysis methods for digital journalism research"; Digital Journalism, 4(1), (2016), 75-88. https://doi.org/10.1080/21670811.2015.1093270

[Hu et al., 2012] Hu, M., Liu, S., Wei, F,, Wu, Y., Stasko, J., Ma, K.: "Breaking news on Twitter"; International ACM SIGCHI Conference on Human Factors in Computing Systems, Austin, TX, USA, (2012), 2751-2754.

[Irfani et al., 2018] Irfani, F. F., Fauzi, M. A., Sari, Y. A.: "News Classification on Twitter Using Naive Bayes and Hypernym-Hyponym Based Feature Expansio"; IEEE International Conference on Sustainable Information Engineering and Technology, Malang, Indonesia, (2018), 317-321.

[Jackoway et al., 2011] Jackoway, A., Samet, H., Sankaranarayanan, J.: "Identification of live news events using Twitter"; International ACM Workshop on Location-Based Social Networks, Chicago, Illinois, USA, (2011), 25-32.

[Jang et al., 2019] Jang, B., Kim, I., Kim, J. W.: Word2vec convolutional neural networks for classification of news articles and tweets. PloS one, 14(8), (2019), e0220976.

[Jiao et al., 2020] Jiao, P., Veiga, A., Walther, A.: "Social media, news media and the stock market"; Journal of Economic Behavior & Organization, 176, (2020), 63-90. https://doi.org/10.1016/j.jebo.2020.03.002

[Juba and Le, 2019] Juba, B., Le, H. S.: "Precision-recall versus accuracy and the role of large data sets"; AAAI Conference on Artificial Intelligence, 33, 01, (2019), 4039-4048.

[Kabra and Shrawne, 2020] Kabra, A., Shrawne, S.: "Location-Wise News Headlines Classification and Sentiment Analysis: A Deep Learning Approach"; International Conference on Intelligent Computing and Smart Communication, Singapore, (2020), 383-391.

[Karadağ and Takçı, 2010] Karadağ, A., Takçı, H.: "Detection Similar News with Text Mining (article in Turkish with an abstract in English)"; (2010), Akademik Bilişim, Muğla, Turkey

[Karayiğit et al., 2021] Karayiğit, H., Acı, Ç. İ., Akdağlı, A.: "Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods"; Expert Systems with Applications, 174, 114802, (2021), https://doi.org/10.1016/j.eswa.2021.114802

[Keskar et al., 2020] Keskar, D., Palwe, S., Gupta, A.: "Fake News Classification on Twitter Using Flume, N-Gram Analysis, and Decision Tree Machine Learning Technique"; International Conference on Computational Science and Applications, Singapore, (2020), 139-147

[Kiliç et al., 2015] Kiliç, E., Tavus, M. R., Karhan, Z.: "Classification of breaking news taken from the online news sites"; 23nd Signal Processing and Communications Applications Conference (SIU), (2015), 363-366.

[Kohavi, 1995] Kohavi, R.: "A study of cross-validation and bootstrap for accuracy estimation and model selection": International ACM Joint Conference on Artificial Intelligence, Montreal, Quebec, Canada, (1995), 1137-1143.

[Kotzé et al., 2020] Kotzé, E., Senekal, B.A., Daelemansi, W.: "Automatic classification of social media reports on violent incidents in South Africa using machine learning"; South African Journal of Science, 116(3-4), (2020), 1-8. https://doi.org/10.17159/sajs.2020/6557

[Kowsari et al., 2019] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: "Text classification algorithms: A survey": Information, 10(4), (2019), 150. https://doi.org/10.3390/info10040150

[Lau, 2014] Lau, C.H.: "Detecting news topics from microblogs using sequential pattern mining"; PhD Dissertation, Queensland University of Technology, Queensland, Australia, (2014.)

[Levenshtein, 1966] Levenshtein, V.I.: "Binary codes capable of correcting deletions, insertions and reversals"; Soviet Physics Doklady, 10, (1966), 707-710.

[Lewis et al., 2019] Lewis, S.C., Guzman, A.L., Schmidt, T.R.: "Automation, journalism, and human–machine communication: Rethinking roles and relationships of humans and machines in news"; Digital Journalism 7(4), (2019), 409-427. https://doi.org/10.1080/21670811.2019.1577147

[Lim et al., 2020] Lim, S.L.O., Lim, H.M., Tan, E.K., Tan, T.P.: "Examining Machine Learning Techniques in Business News Headline Sentiment Analysis"; International Conference on Computational Science and Technology, Singapore, (2020), 363-372

[Liu et al., 2017] Liu, X., Nourbakhsh, A., Li, Q., Shah, S., Martin, R., Duprey, J.: "Reuters tracer: Toward automated news production using large scale social media data"; International IEEE Conference on Big Data (Big Data), Boston, MA, USA, (2017), 1483-1493.

[Mandal and Sen, 2014] Mandal, A. K., Sen, R.: "Supervised learning methods for bangla web document categorization"; International Journal of Artificial Intelligence & Applications, 5(5), (2014), 93-105. https://doi.org/10.5121/ijaia.2014.5508

[Mukherjee and Sarkar, 2020] Mukherjee, S., Sarkar, K.: "Analyzing Large News Corpus Using Text Mining Techniques for Recognizing High Crime Prone Areas"; International IEEE Calcutta Conference, Kolkata, India, (2020), 444-450

[Nergiz et al., 2019] Nergiz, G., Safali, Y., Avaroğlu, E., Erdoğan, S.: "Classification of Turkish news content by deep learning based LSTM using Fasttext model"; IEEE International Artificial Intelligence and Data Processing Symposium, Malatya, Turkey, (2019), 1-6

[Nourbakhsh et al., 2017] Nourbakhsh, A., Li, Q., Liu, X., Shah, S.: “"Breaking" Disasters: Predicting and Characterizing the Global News Value of Natural and Man-made Disasters”, (2017), arXiv preprint arXiv:1709.02510.

[Obiedat, 2020] Obiedat, R.: “Predicting The Popularity of Online News Using Classification Methods With Feature Filtering Techniques”; Journal of Theoretical and Applied Information Technology, (2020), 98(08)

[Özbay and Alataş, 2020] Ozbay, F.A., Alatas, B.: “Fake news detection within online social media using supervised artificial intelligence algorithms”; Physica A: Statistical Mechanics and its Applications, 540, (2020), 123174. https://doi.org/10.1016/j.physa.2019.123174

[Sathyadevan and Nair, 2015] Sathyadevan, S., Nair, R.R.: “Comparative analysis of decision tree algorithms: ID3, C4. 5 and random forest”; Computational intelligence in data mining, 1, (2015), 549-562. https://doi.org/10.1007/978-81-322-2205-7_51

[Schwartz et al., 2015] Schwartz, R,, Naaman, M., Teodoro, R.: “Editorial Algorithms: Using Social Media to Discover and Report Local News”; International AAAI Conference on Web and Social Media, Oxford, UK, (2015), 407-415.

[Shah et al., 2020] Shah, K., Patel, H., Sanghvi, D., Shah, M.: “A comparative analysis of logistic regression, random forest and KNN models for the text Classification”; Augmented Human Research, 5(1), (2020), 1-16.

[Shu et al., 2018] Shu, K., Wang, S., Liu, H.: “Understanding user profiles on social media for fake news detection”; IEEE International Conference on Multimedia Information Processing and Retrieval, Miami, Florida, USA, (2018), 430-435

[Singh et al., 2020] Singh, R., Chun, S.A., Atluri, V.: “Developing Machine Learning Models to Automate News Classification”; 21st Annual International Conference on Digital Government Research, Seoul, Korea, (2020), 354-355

[Song et al., 2014] Song, G., Ye, Y., Du, X., Huang, X., Bie, S.: “Short text classification: A survey. Journal of multimedia”; (2014), 9(5), 635.

[Şen and Yanıkoğlu, 2018] Şen, M. U., Yanıkoğlu, B.: “Classification of SuDer Turkish news corpora”; IEEE 26th Signal Processing and Communications Applications Conference, İzmir, Turkey, (2018), 1-4.

[Toraman et al., 2011] Toraman, C., Can, F., Kocberber, S.: “Developing a text categorization template for Turkish news portals”; International IEEE Symposium on Innovations in Intelligent Systems and Applications, İstanbul, Turkey, (2011), 379-383

[Trieu et al., 2017] Trieu, L. Q., Tran, H. Q., Tran, M. T.: “News classification from social media using twitter-based doc2vec model and automatic query expansion”; Eighth International Symposium on Information and Communication Technology, Nha Trang, Vietnam, (2017). 460-467.

[Tüfekci et al., 2012] Tüfekci, P., Uzun, E., Sevinç, B.: “Text classification of web based news articles by using Turkish grammatical features”; International IEEE Signal Processing and Communications Applications Conference, Muğla, Turkey, (2012) , 1-4

[Twitter Developers, 2021] Twitter Developers: “Docs-Twitter Developers”, https://dev.twitter.com/streaming/overview, Accessed 28 May 2021.

[Uslu and Özmen Akyol, 2021] Uslu, O., Özmen Akyol, , S.: “Türkçe Haber Metinlerinin Makine Öğrenmesi Yöntemleri Kullanılarak Sınıflandırılması”; Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi, 2(1), (2021), 15-20.

[Uysal and Gunal, 2014] Uysal, A.K., Gunal, S.: "The impact of preprocessing on text Classification"; Information Processing & Management, 50(1), (2014), 104-112. https://doi.org/10.1016/j.ipm.2013.08.006

[Vaseeharan and Aponso, 2020] Vaseeharan, T., Aponso, A.: "Review On Sentiment Analysis of Twitter Posts About News Headlines Using Machine Learning Approaches and Naïve Bayes Classifier"; 12th International Conference on Computer and Automation Engineering, (2020), 33-37

[Vermeer et al., 2020] Vermeer, S., Trilling, D., Kruikemeier, S., de Vreese, C.: "Online News User Journeys: The Role of Social Media, News Websites, and Topics"; Digital Journalism, (2020), 1-28. https://doi.org/10.1080/21670811.2020.1767509

[Yang et al., 2019] Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., Liu, H.: "Unsupervised fake news detection on social media: A generative approach"; AAAI International Conference on Artificial Intelligence, Honolulu, Hawaii, USA, (2019), 5644-5651

[Yolcu and Demirci, 2021] Yolcu, A. B., Demirci, S.: "Twitter'daki Türkçe Haber Paylaşımlarının Derin Öğrenme Tabanlı Sınıflandırılması"; IEEE International Conference on Computer Science and Engineering, Ankara, Turkey, (2021).

[Zaman et al., 2020] Zaman, B., Justitia, A., Sani, K.N., Purwanti, E.: "An Indonesian Hoax News Detection System Using Reader Feedback and Naïve Bayes Algorithm"; Cybernetics and Information Technologies, 20(1), (2020), 82-94. https://doi.org/10.2478/cait-2020-0006