Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
Anatolian Science

ISSN,e-ISSN: 2548-1304
Volume:9, Issue:2, pp:186-201, 2024
https://doi.org/10.53070/bbd.1545101
Research Paper

Optimizing Turkish Opinion Mining: A Comparative Study of AI Algorithms

Türkçe Görüş Madenciliğinin Optimize Edilmesi: Yapay Zekâ Algoritmalarının Karşılaştırmalı Bir Çalışması

Ömer Köksal*[1]

[1]Data Science and Artificial Intelligence Department, University of Doha for Science and Technology, Doha, Qatar

(omer.koksal@udst.edu.qa)

Received:Sep.07, 2024 Accepted:Dec.21, 2024 Published:Dec.25, 2024

Özetçe—Fikir madenciliği ya da diğer adıyla duygu analizi, metin verilerinde ifade edilen görüşleri, duyguları, tutumları ve hisleri analiz etmeye ve anlamaya odaklanan Doğal Dil İşlemenin (NLP) bir dalıdır. Fikir madenciliğinin amacı, bir inceleme, yorum veya sosyal medya gönderisi gibi belirli bir metin parçasının duygu kutupluluğunu belirlemektir. Ancak görüş madenciliği, daha az araştırılmış dillerdeki çalışmaları İngilizce yapılan çalışmalardan ayıran dile özgü zorluklarla karşı karşıyadır. Bu makale, çeşitli yapay zekâ algoritmalarını karşılaştırarak Türkçe fikir madenciliği için yeni bir süreç sunmaktadır. Şeffaflık ve yeniden üretilebilirliği sağlamak için açık kaynaklı bir Türkçe görüş madenciliği veri kümesi kullanarak kapsamlı deneyler yürüttük. Araştırmamızda geleneksel makine öğrenimi, derin öğrenmeye dayalı algoritmalar ve önceden eğitilmiş dönüştürücü modelleri değerlendirerek parametrelerini optimize etmeye odaklandık. Ayrıca kelime yerleştirmelerini geleneksel kelime torbası yöntemiyle karşılaştırdık. Hiper parametreler ince ayarlanması ile optimize edilmiş modellerimiz sınıflandırma performansını önemli ölçüde iyileştirdi. Önerilen süreç, literatürdeki çalışmalarda %80,1'den başlayan doğruluk seviyesini %97,19'a çıkartarak fikir madenciliğindeki gelecekteki araştırmalar için değerli içgörüler sağlamıştır.

Anahtar Kelimeler: *Fikir madenciliği, doğal dil işleme, makine ögrenmesi, derin ögrenme, ön-eğitimli dil modelleri, dönüştürücü algoritmaları.*

Abstract— Opinion mining, aka sentiment analysis, is a branch of Natural Language Processing (NLP) that focuses on analyzing and understanding opinions, sentiments, attitudes, and emotions expressed in text data. The goal of opinion mining is to determine the sentiment polarity of a given piece of text, such as a review, comment, or social media post. However, opinion mining faces language-specific challenges that differentiate studies in less commonly researched languages from those conducted in English. This article presents a novel process for Turkish opinion mining by comparing various artificial intelligence algorithms. We conducted extensive experiments using an open-source Turkish opinion-mining dataset to ensure transparency and reproducibility. Our research evaluated traditional machine learning, deep learning-based algorithms, and pre-trained transformer models, focusing on optimizing their parameters. We also compared word embeddings with the traditional bag-of-words method. By fine-tuning hyperparameters, our optimized models significantly improved the classification performance. The proposed process improved the accuracy level from 80.1% in literature studies to 97.19%, providing valuable insights for future research in opinion mining.

Keywords: *Opinion mining, natural language processing, machine learning, deep learning, pre-trained language models, transformer models.*

1. Introduction

Opinion mining is a widely used task in natural language processing, particularly in the realm of text classification. Its purpose is to learn about customers' opinions on a product or service, which can be used to improve it. This involves collecting customer opinions and classifying them as either positive or negative. Some studies even add a "neutral" class or break opinions into five categories. Traditional machine learning algorithms are commonly used for classification, but deep learning algorithms and pre-trained language model-based transformer algorithms have shown great success in recent years. Overall, opinion mining is crucial for businesses looking to understand their customers better and improve their products or services. Opinion mining, like almost all NLP tasks, is language-dependent. In other words, the success of the algorithms and techniques used may vary depending on the language being studied and the structure of this language. However, most of the NLP studies in the literature are used on English datasets, which is the most common language, and some algorithms and processes are developed primarily for this language. Turkish, unlike English, is an agglutinative language, and its grammatical structure differs from English. Therefore, it cannot be guaranteed that an algorithm that gives successful results in English will also be successful in other languages.

While many NLP studies have been conducted, only some have compared various algorithms and techniques for Turkish opinion mining. This article addresses this gap by comparing opinion-mining experiments' results using different learning-based classification algorithms. Furthermore, the article proposes a new process for optimizing the hyperparameters in these algorithms and evaluates their impact on the results.

An open-source Turkish opinion-mining dataset was used to conduct these experiments, previously utilized in several pieces of literature research. The study conducted extensive experiments on this dataset using four main categories of artificial intelligence algorithms to demonstrate their comparative effects on opinion mining. First, traditional machine learning algorithms were used, including Naive Bayes (NB), Support Vector Machines (SVM), Logistic Regression (LR), K-Nearest Neighbor (KNN), and the Decision Tree (DT) Algorithm. The Random Forest (RF) Algorithm, an advanced ensemble learning method, was also employed. In the second stage, experiments were carried out with Deep Learning-based algorithms, namely deep neural networks (DNN) and convolutional neural networks (CNN). In stages one and two, feature vectors were obtained using the traditional bag of words and word embedding methods. In the final stage, pre-trained language models (PLM) based on Transformer Algorithms were tested on the same data set. Multiple PLM models were utilized in these experiments, and their effects on the results were examined.

The article is organized into several sections, each focusing on a specific aspect of the research. Section 2 highlights relevant studies, while Section 3 summarizes algorithms and techniques discussed in the article. Section 4 delves into the dataset used in the experiments, and Section 5 explains the experimental setup and presents the results obtained. Section 6 evaluates the experimental results and compares them with existing studies in the literature. Finally, Section 7 concludes the paper.

2. Related Work

In the literature, there are different studies on Turkish opinion mining. Most of these studies include opinion-mining studies using social media data and customer feedback about company products. When these studies mainly used word-based approaches and traditional Machine machine-learning algorithms. In recent studies, PLM-based opinion mining studies, word representations, and Deep Learning Algorithms are also included in the literature. These studies used the accuracy and F1 measure metrics as the evaluation criteria. In this section, some opinion mining studies in the literature will be given in two parts. Firstly, general Turkish opinion mining studies will be discussed, and then the studies using the data set used in this study will be mentioned.

2.1. The research in the Turkish Opinion Mining

In this section, some research on Turkish opinion mining will be discussed.

Rumelli et al. (2019) used Machine Learning Algorithms in opinion mining using dictionary-based approaches in their study. Using KNN, NB, RF, and SVM Algorithms, they classified customer opinions and product comments received from an electronic commerce site into three headings (positive, negative, neutral), using the SentiTürk (Dehkharghani, 2016) library developed for Turkish and reached an accuracy rate of 73%.

Çiftçi and Apaydın (2019) conducted a Turkish opinion-mining study with Deep Learning Algorithms. In their study, customer comments received from e-commerce and movie review sites were classified with an accuracy of 83.3% using the Long Short-Term Memory (LSTM) Algorithm. The authors stated they achieved 3.2% and 0.6% higher success rates than the NB and LR classifiers used in this article.

Açıkalın, Bardak, and Kutlu (2020) used pre-trained language models in Turkish opinion mining. In this study, where BERT was used as a pre-trained language model, hotel and movie reviews were automatically translated from English to Turkish. It was stated that the accuracy value achieved in this opinion mining study conducted with two classes was 93.32%.

2.2. The research in the Turkish Opinion Mining that used the Demirtaş Dataset

This section will mention studies such as this study using the open-source Demirtaş and Pechenizkiy (2013) Turkish opinion mining dataset. The original data set includes product and movie reviews in English and Turkish.

By sharing this data set as open source, it has been used in many opinion-mining studies in the literature and has become a benchmark.

In their study, Demirtaş and Pechenizkiy (2013) used English data sets obtained using automatic machine translation and opinion mining to increase the accuracy rate of translations made in this way. The authors researched opinion mining studies in Turkish. Then, they examined the English movie and product training set obtained through machine translation using NB, SVM, and Maximum Entropy (ME) algorithms in their research. They increased the accuracy rate from 69.5% to 80.10% on the film data set with the NB Algorithm.

Gözükara and Özel (2016) examined the effects of different Vectorization techniques used in opinion mining in detail. For this purpose, they used different preprocessing, normalization, and bag-of-words [BOW] methods in their experiments. As a result of these studies, the authors reported that they achieved an accuracy rate of 88.21% with the SVM classifier.

Kurt, Kısa, and Karagöz (2019) examined segmentation approaches in opinion mining research consisting of short texts. After using different preprocessing techniques, they tried to increase the accuracy rate they achieved by using two Deep Learning Algorithms: convolutional neural networks and recurrent neural networks. The authors stated that in this study, they achieved an accuracy rate of 90.80% using the LSTM Algorithm.

Görmez et al. (2020) proposed a new method for Turkish opinion mining. Using three different data sets, they performed opinion mining with the "Feature-based stacked group method" (FBSEM) and Machine Learning classifiers. They reached 89.1% accuracy in the Demirtaş data set, one of the Turkish data sets they used.

Yıldırım (2020) compared Deep Learning Algorithms and bag-of-words processes in opinion mining. CNN and RNN-based techniques were used as Deep Learning Algorithms. Demirtaş reported that they achieved an accuracy value of 89.26% in the movie data set by combining LSTM and dilution algorithms.

3. Algorithms and Techniques Used in Opinion Mining

In this section, Algorithms and techniques frequently used in opinion mining will be briefly mentioned. Since opinion mining is considered a sub-branch of text classification, text classification, preprocessing, and vectorization of texts will be discussed first. Then, brief information about classification algorithms will be given. Algorithms will be described in three subsections: traditional Machine Learning Algorithms, deep learning Algorithms, and transformer Algorithms-based pre-trained language models. Finally, the metrics used to evaluate our classification models will be mentioned.

3.1. Text Classification

Text classification can be defined as dividing documents into different categories according to the information they contain. Classifications made in opinion mining are generally made into two categories: positive and negative. However, a small number of three-class (positive, neutral, and negative) and five-class (extreme negative, negative, normal, positive, and extreme positive) studies are also carried out. Text classification can be performed using learning-based, rule-based, or hybrid methods. In this study, learning-based Algorithms and techniques will be discussed.

3.2. Preprocessing

Text data preprocessing transforms raw input text data into new, more suitable representations for text classification. In many text classification applications, intentional or unintentional errors might exist in the input text data. The customers' feedback is generally unstructured, and there might be comments containing a few misspelled words and deliberate errors, such as 'spr' or 'supperrr' instead of 'super'. In comments, Turkish-specific letters (ö, ü, ş, ç, ğ, ı) are used, and sometimes words are written by adding English characters (x, w, q) deliberately. Even if spell checkers are used in applications, these errors should be corrected in a context-sensitive manner not to cause a loss of meaning. These corrections made with preprocessing can help increase classification accuracy by reducing the feature sizes used. Different techniques are used individually or together in the preprocessing step. In the following sections, these techniques will be briefly mentioned.

Tokenization is the division of the input text into smaller parts. During the tokenization process, punctuation marks, delimiters, extra spaces, and emoticons are removed from the text data.

Lowercase conversion converts all input text to lowercase, which is one of the most commonly used techniques in the preprocessing phase. This transformation reduces feature size and improves classification accuracy in almost all cases (Işık, 2020).

Stop word removal is a widely used technique in text preprocessing. The most frequently used words in a particular language are called stop words. Stop words are assumed to be irrelevant within the study domain and to have no impact on classification. With this process, stop words are filtered out before the classification stage. Thus, removing stop words also reduces the size of the run.

Stemming is obtaining an inflected word's root or root form by removing the suffix. It is a language-specific process that provides a standard form for variants of words. Therefore, word frequencies are typically counted after stem extraction in the Vectorization process.

The *lemmatizing* process results in obtaining the semantic root of an inflected word based on its intended meaning. Moreover, while the semantic stemming process considers the word's morphological information, stemming directly cuts off the end or beginning of a word to obtain its stem.

3.3. Vectorization of the Text Input

In text classification, text data must first be converted into numerical formats. The process of converting text data into vector space models is called Vectorization. The traditional "bag of words" method and the newer "word embeddings" technique are widely used for text vectorization.

Vectorization with "Bag of Words" is a characteristic technique used in traditional text classification applications, where text vectors are obtained by working on existing text data. In this model, input text data are represented as vectors, and all terms correspond to features. In other words, different words in the documents determine the size of the feature space. This model preserves the word order in the text and distinguishes the presence of a word in the document. The weights of the extracted features are determined using different methods, such as the bag-of-words approach (Gomes, 2019).

The term frequency (TF) method considers the frequencies of terms appearing in each text. Thus, the TF Equation can be expressed in the following equation with the parameters t (term) and d (document) as given below:

$$\mathrm{TF}\ (\mathrm{t},\mathrm{d})\ =\ \log\ (1+\mathrm{freq}(\mathrm{t},\mathrm{d})) \qquad (1)$$

In the Term Frequency – Inverse Document Frequency (TF-IDF) method, the weights are calculated by considering the frequencies in all collections. The importance of terms is related to their frequency in the relevant document and is inversely proportional to the frequency of the term in the collection. TF-IDF can be expressed as given in the following equation:

$$\mathrm{TFIDF}\ (\mathrm{t},\mathrm{d},\mathrm{D})\ =\ \mathrm{TF}(\mathrm{t},\mathrm{d})\ \mathrm{x}\ \mathrm{IDF}(\mathrm{t},\mathrm{D})) \qquad (2)$$

The definition of the IDF term in this equation is given below.

$$\mathrm{IDF}\ (\mathrm{t},\mathrm{D})\ =\ \log\ (\mathrm{N}\ /\ \mathrm{count}\ (\mathrm{d}€\mathrm{D}\colon \mathrm{t}€\mathrm{d})) \qquad (3)$$

In the "Word Embeddings" technique, vectors are obtained from the input texts and much larger text data in that language (such as internet text data). Word representations work with the mathematical vectors that words represent, and more precise results can be obtained from bag-of-word models. Word representations solve the problem of data sparsity in the bag-of-words method by considering the words in the neighborhood of the focused word. In this way, words with similar meanings can be represented close to each other in the vector space.

3.4. Traditional Machine Learning Algorithms

This subsection briefly explains the traditional machine learning algorithms used in this study.

Naive Bayes (NB) is widely used as a probabilistic classifier in classification tasks. Since NB has a low calculation time, it can quickly give an opinion of the data set used. NB is based on the Bayes Theorem, which assumes that a given feature value does not depend on other features (Alpaydın, 2016). This assumption is only sometimes valid in real life. Different NB classifiers have been developed to overcome the limitation of this independence assumption. Polynomial NB, Complement NB, and Bernoulli NB are well-known NB-based Algorithms developed for this purpose.

Logistic Regression (LR) is a frequently used classification algorithm. Although its mathematical definition is similar to linear regression, LR classifies by comparing the output of the logistic function with a threshold value determined depending on the problem domain (Köksal, 2021).

Support Vector Machine is a widely used classifier in text classification applications as it outperforms other Machine Learning Algorithms in many applications. SVM creates optimal hyperplanes, called decision surfaces, to classify data. SVM iteratively uses these hyperplanes to minimize the error. A hyperplane is drawn between support vectors belonging to different categories. The data points closest to the hyperplanes are called support vectors. SVM uses several kernels to create hyperplanes. Linear (SVM-l), polynomial (SVM-p), radial basis function (SVM-rbf), and sigmoid (SVM-s) are the most commonly used SVM kernels. Depending on the data type and size, the kernel can achieve higher classification accuracy (McMahan, 2019; Köksal, 2022). Therefore, SVM can be used instead of Deep Learning-based classifiers for small data as it provides similar performance with less training time.

The K-Nearest Neighbors Algorithm is based on feature similarity and does not make any assumptions for data distribution. KNN makes predictions based on neighboring data points. 'K' is the only input parameter determining how much data the Algorithm will classify. The fast-training phase of this Algorithm provides an advantage, especially when processing data without prior knowledge.

Decision Tree is a predictive modeling approach that uses a tree structure to break complex data into more manageable pieces. In this structure, internal nodes correspond to tests on attributes, the branches represent the test results, and leaf nodes represent class labels. Classification in the Decision Tree is performed iteratively by starting at the root node, testing the node's attributes, and moving down based on the test results.

Random Forest algorithm performs classification and regression tasks using many decision trees simultaneously, giving better results than decision trees. Hence, the algorithm generates different models on the same data set and a separate Decision Tree for each sample. The results obtained are evaluated by voting, and the result with the most votes is selected.

3.5. Deep Learning-based Classification Algorithms

Inspired by the human visual system, the CNN Algorithm uses convolution operations to obtain input features. In the convolution process, the result is obtained by multiplying matrices using multiple sliding filters with the image data. This process provides access to different information about the input data. CNN is also widely used in fields other than vision. For example, 1D convolution operations can be applied in various fields, such as examining sequential data, text, or time series data to find patterns in data (McMahan, 2019). In video processing, 3D convolution processes the first two dimensions to form a frame, whereas the other dimension represents a time series.

3.6. Transformer Algorithm-based Pre-Trained Language Models

In recent years, the best results in many NLP tasks have been achieved using pre-trained language models. Due to their successful results, the research conducted in this field has increased significantly with the introduction of transformer algorithms (Vaswani, 2017).

In these algorithms, which also use encoder and decoder models, input texts are first converted into high-dimensional hidden embeddings by the encoder. The dynamic representations obtained from these embeddings are translated back into natural language format with the solver. An attention mechanism is used to model long-distance dependencies in the input text. In these models, in addition to obtaining field-independent large corpora through the pre-training process, it is also possible to fine-tune by using small, unique data of the field/data set. To better examine dependencies, bidirectional models aim to capture contextual information about the prediction process. BERT (Bidirectional Encoder Representations from Transformers), one of the most used bidirectional coding models, is one of the most used models in this field (Devlin, 2019). With the prediction of the following sentence and masked language modeling techniques, many studies in the literature have shown that BERT models used in different natural language processing tasks achieve more successful results compared to other pre-trained language models (Devlin, 2019; Ambalavanan, 2020; Köksal, 2021; Köksal, 2022).

Efficiently learning an encoder that classifies token replacements accurately (ELECTRA) is another widely used pre-trained language model based on the Transformer Algorithm (Clark, 2020). The ELECTRA uses an architecture similar to the Generative Adversarial Network (GAN).

3.7. Evaluation Metrics

In classification tasks, most evaluation metrics are based on the confusion matrix, which reveals how our model is confounded across prediction classes. Although the confusion matrix is a two-by-two matrix in binary classification, its usage can be extended for multi-class classification tasks (Köksal, 2022). The confusion matrix shows the true and false positives and negatives. From these values, many metrics can be obtained to evaluate the model. This paper considers our models using the accuracy metric based on the confusion matrix to be comparable with the previous studies conducted on the same dataset.

4. Opinion Mining Dataset

The experiments in this article were performed on the Demirtaş film dataset (Demirtaş, 2013), first published in the Demirtaş and Pechenizkiy article in 2013. The authors have made this data set open access, and it has been used in many Turkish opinion mining studies in the literature. Details of the dataset used are given in Table 1.

Table 1. Details of the Demirtas movie dataset

Opinions	Instance
Positive (+)	5331
Negative (-)	5331

Previous studies conducted with this data set are summarized above in the "Related Studies" section, and their results are presented. Additionally, our experimental results in this article are compared with studies using this data set in the literature in the "Evaluation" section.

5. Methodology

This section explains the opinion-mining process proposed in the article and identifies the research questions.

5.1. The Proposed Opinion-Mining Process

This section describes a new opinion-mining process used in the study. The main components of the opinion-mining process are given in Figure 1 in the Business Process Management Notation (BPMN) (Chinosi, 2012; OMG, 2024).

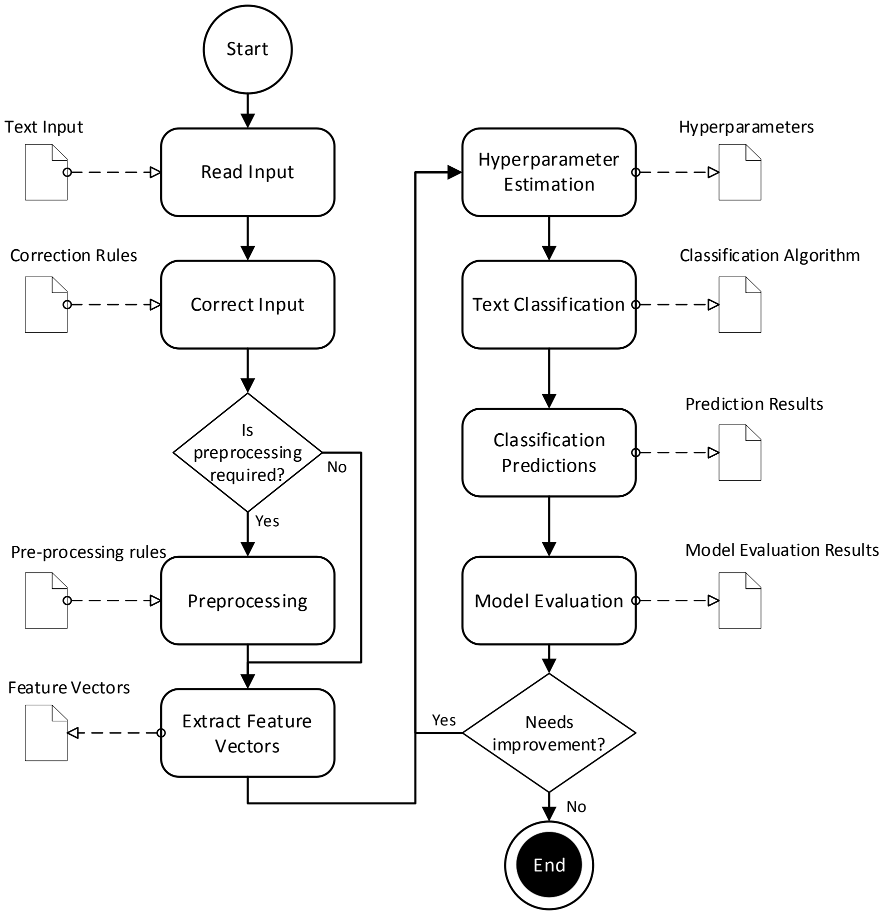

Figure 1. The steps of the proposed opinion mining process

The process starts with reading the text data, as shown in Figure 1. Next, the text data is checked for missing or incorrect data. If so, these data will be corrected or may not be evaluated. Afterward, preprocessing techniques can be applied. Preprocessing techniques to be used in a particular opinion-mining should be performed in Turkish. Preprocessing, such as removing stop words and going to the semantic root, should be done with libraries that support this language. After preprocessing processes, text data must be digitized, that is, vectorized. The two most used methods for vectorization (bag of words and word representations) were used separately in this study. Also, their effects on the results were reported. The next stage is determining the Hyperparameters used in the text classification Algorithms. Hyperparameters are parameters that artificial intelligence algorithms cannot learn and are generally assigned manually. Since these parameters significantly affect the model's performance, it is necessary to improve the model evaluation results by optimizing them. Hyperparameters are typically optimized based on previous experience or by iteratively observing the effects of different values on the result. This step involves long iterations. The specified hyperparameters and the selected algorithm are used as inputs for the classification process. Algorithm selection is also a choice based on experience, and different Algorithms can give better results depending on the structure of the data set. By training the text classification model, class predictions are made for the data, and the success of the resulting model is evaluated based on the selected metrics.

In this study, the proposed opinion mining process was used with different algorithms, and the optimized Hyperparameters according to the data set were obtained with long periods and many iterations. The following sections give the experimental results obtained, the evaluations of these results, and the process's success.

6. Research Questions

This study aims to answer the following three research questions (RQ) identified:

- **RQ.1** What algorithms provide better results in opinion mining?
- **RQ.2** Can more successful results be achieved in opinion mining using word embeddings?
- **RQ.3** Are pre-trained language models based on the transformer algorithm increasing success with opinion mining?

The answers to the research questions given above were revealed through experiments and evaluated in the "Discussion" section.

7. Experiments and Results

This section presents the different experiments performed and experimental setups used in the study. Then, the acquired results will be provided.

7.1. Experimental Setup

In experiments conducted with traditional machine learning, a 10-fold cross-validation technique was used. The experiments were run on a computer with a 24-core dual Xeon processor, four GPUs, and 64 GB memory.

7.2. The Results of Machine Learning Algorithms with TF-IDF

Table 2. The results of the machine learning algorithms

No	Algorithm	Vectorization	Accuracy [%]
1	Naïve Bayes	TF-IDF	83.28
2	SVM-linear	TF-IDF	89.48
3	SVM-poly	TF-IDF	88.41
4	**SVM-rbf**	**TF-IDF**	**89.77**
5	SVM-sigmoid	TF-IDF	89.40
6	KNN	TF-IDF	87.75
7	Logistic Regression	TF-IDF	89.10
8	Decision Tree	TF-IDF	79.12
9	Random Forest	TF-IDF	87.45

In this section, six traditional machine learning algorithms frequently used in text classification and opinion mining are used in the experiments. Four different cores of the SVM were used for the SVM algorithm, and the total number of algorithms was determined to be nine. Unlike other studies, we first tried to find the best word bag size. The bag of words technique, which has a size of 10,000, provides the best results. The variation of the Algorithm results used compared to BOW and the results obtained are given in Table 2. The results are for the BOW 10000 value, as stated above.

Besides the best results presented in Table 2, the results of Machine Learning Algorithms are provided in Figure 2 for different BOW values.

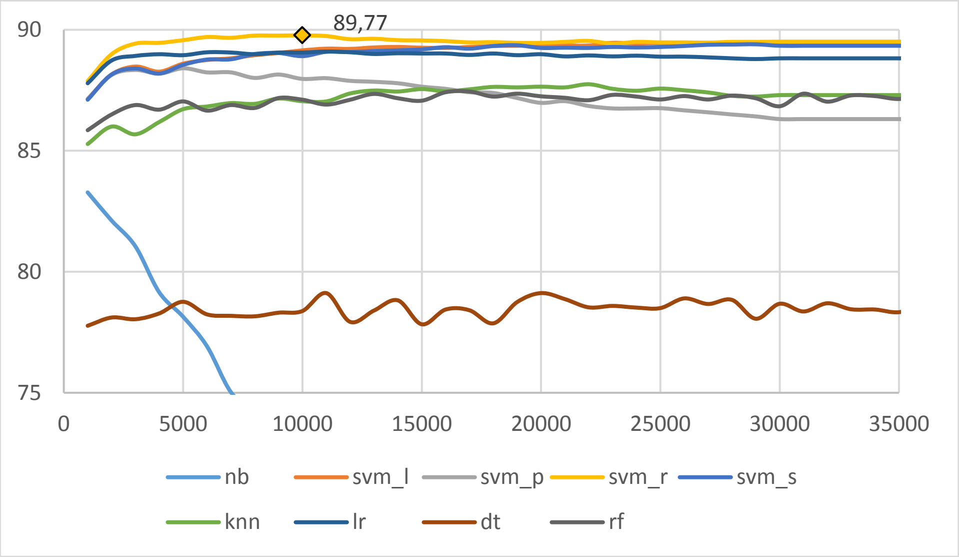

Figure 2. The variation of the machine learning algorithm's results with BOW size

The best results obtained with the three Algorithms in which the best results were obtained are presented in Figure 3. As seen in Table 2, the best results in this section were obtained with the SVM Algorithm RBF kernel. When the word bag size exceeds 35K, all Algorithms except two produce fixed values, and the results no longer change with the word bag size.

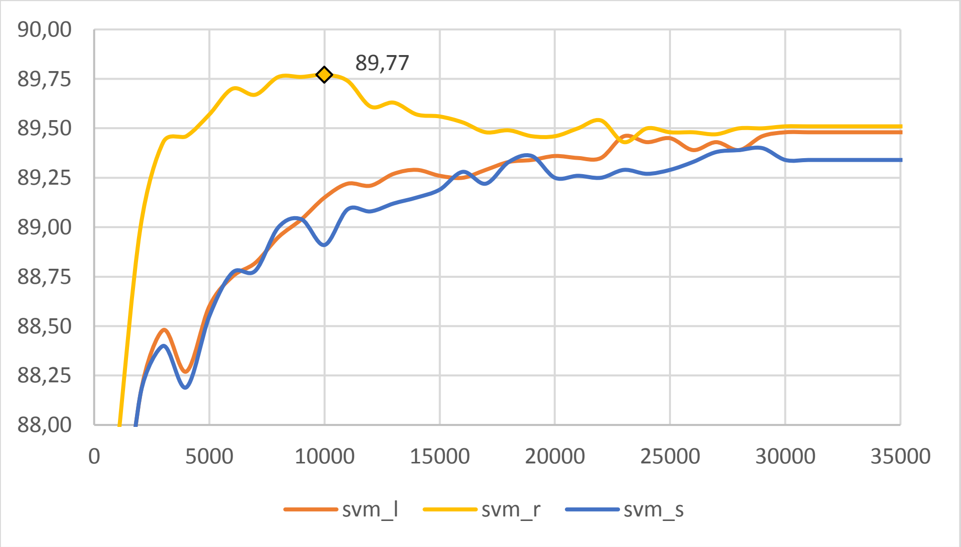

Figure 3. The variation of the best three machine learning algorithm results with BOW size

These results show that all the information within the dataset, with a size of 35000, was exploited in the bag of words technique. Although only decision tree and random forest algorithms produce different results due to their random prediction feature in their structure, the variations of these results are minor compared to other intervals.

7.3. The Results Obtained with N-Gram Technique

In this section, the results obtained with different N-Gram techniques are given. Different N-Gram combinations were used in the SVM-rbf Algorithm, where we obtained the best results. N (1, 1) shows the feature set obtained by using only one word, and N (2, 2) shows the feature set obtained by using two consecutive words. N (1, 2) refers to the feature set obtained by using both one and two words. All combinations were tried up to the feature set obtained with four consecutive words, and the results are given in Table 3.

Table 3. Accuracy values obtained with N-Gram and SVM-rbf

No	N-Gram Method	Accuracy [%]
1	(1, 1)	89.77
2	**(1, 2)**	**90.32**
3	(2, 2)	86.11
4	(3, 1)	90.29
5	(3, 2)	85.92
6	(3, 3)	79.30
7	(1, 4)	90.29
8	(2, 4)	85.64
9	(3, 4)	78.91
10	(4, 4)	63.38

7.4. The Results of the Machine Learning Algorithms with Word Embeddings

In Section 4.1, Vectorization was performed using the bag-of-words method while using Machine Learning-based Algorithms. In this section, Vectorization is carried out with word representation libraries. It is seen in the literature that there are different word representation libraries, such as Word2Vec (Mikolov, 2013), Doc2Vec (Le, 2014), GloVe (Pennington, 2014), and FastText (Bojanowski, 2017). In this study, FastText was used as the word representation library. FastText treats each word as n-gram characters instead of Word2Vec. Therefore, for rare words, FastText provides better word embeddings. It can also create a vector for a word outside the vocabulary that is not present in the training corpus. This feature is not available in Word2Vec or GloVe. The results obtained with FastText with Machine Learning Algorithms are given in Table 4.

Table 4. Accuracy values obtained with word embeddings and machine learning

No	Algorithm	Vectorization	Accuracy [%]
1	Naïve Bayes	FastText	75.26
2	SVM-linear	FastText	85.18
3	SVM-poly	FastText	86.82
4	**SVM-rbf**	**FastText**	**86.96**
5	SVM-sigmoid	FastText	71.00
6	KNN	FastText	77.97
7	LR	FastText	84.34
8	Decision Tree	FastText	69.22
9	Random Forest	FastText	82.31

7.5. The Results of the Deep Learning Algorithms

In this section, experiments were carried out using two selected deep learning algorithms (DNN and CNN), utilizing both vectorization methods given above. The results are presented in Table 5.

Table 5. The results of the Deep Learning algorithms

No	Algorithm	Vectorization	Accuracy [%]
1	DNN	TF-IDF	88.19
2	DNN	FastText	89.27
3	CNN	TF-IDF	81.10
4	**CNN**	**FastText**	**90.84**

7.6. The Results of the Pre-trained Language Models

7.6.1. Selection of the BERT models to be used

Many models developed based on BERT were shared as open source. However, the BERT models most suitable for the NLP tasks should be selected and used in experiments. For example, there are BERT-based models developed in many languages for different NLP tasks, such as text classification, text generation, sentiment analysis, and question answering. For this reason, Turkish opinion mining models were selected and used for this study. However, to see the performance of these models and their contribution to the NLP task, different base models and models particularly trained for this language were selected and used in the experiments.

7.6.2. The BERT models used

Table 6 gives the names and classifications of the 13 BERT models used in the experiments. Below are explanations of these models.

Table 6. The BERT models used

No	Model Name	Abb.	Model Grub
1	bert-base-uncased	BBU	Base models Trained in English
2	bert-base-multilingual-uncased	BBMU	
3	distilbert-base-multilingual-cased	DBMC	
4	roberta-base	RB	
5	dbmdz/bert-base-turkish-uncased	DBTU	Models Trained in Turkish
6	dbmdz/bert-base-turkish-cased	DBTC	
7	dbmdz/bert-base-turkish-128k-uncased	D128U	
8	dbmdz/distilbert-base-turkish-cased	D128C	
9	loodos/bert-base-turkish-cased	LBTC	
10	gurkan08/turkish-product-comment-sentiment-classification	GTSC	Models Trained for Turkish Opinion Mining
11	kuzgunlar/electra- turkish-sentiment-analysis	KETSA	
12	scoup123/autotrain-turkish_sentiment_analysis	STSA	
13	savasy/bert-turkish-sentiment-cased	STSC	

BBU model is one of the native BERT models. As stated in its name, it is a model that works without distinguishing between uppercase and lowercase letters. Experiments conducted with this model (Işık, 2020), which only has English language support, were used in this study to produce fundamental results. BBMU is a native BERT model that provides multi-language support. This model is trained in 112 languages (including

Turkish). It was included in the study to see the differences between this model and BERT models specially trained for Turkish. The DBMC model was similarly trained for 104 different languages, including Turkish. However, the primary purpose of the Distilbert models is to obtain results that are close to the accuracy of the original BERT models, with smaller models and shorter training times (Vaswani, 2017). The "dbmdz/distilbert-base-turkish-cased" model is obtained by training the Distilbert architecture on Turkish data sets. The RB model is the base model of the widely used RoBERTa models. RoBERTa models are BERT-based models that can achieve better results by optimizing BERT models.

DBTU and DBTC models (also called BERTurk) are specifically trained for the Turkish language and frequently used in Turkish SLP tasks. The difference between these two models is whether to consider lower-case letters. The larger models, D128U and D128C, are trained for Turkish with a 128K corpus. Since they are trained with larger data sets, they can provide better results than BERTurk in many classification tasks. LBTC, a similar BERT-based Turkish classification model, was also examined in this study.

This study uses four pre-trained language models specifically for the Turkish opinion-mining task. The GTSC, STSA, and STSC models are trained especially for Turkish sentiment analysis. The KETSA model, also trained for the Turkish opinion-mining task, is ELECTRA-based.

7.6.3. Best Hyperparameter Values Used in The Bert Models

In the selected BERT models, long iterations were carried out to obtain better results, and the Hyperparameters used were tried to be optimized. Optimization of these hyperparameters, which make significant differences in the results, was one of the most critical parts of this study. The most optimized Hyperparameter values are given in Table 7.

Table 7. The Tuned Deep Learning Hyperparameters

Hyperparameter	CNN	DNN
Learning rate	1.0 E-5	1.0 E-5
Number of epochs	30	30
Batch size	32	16
Optimization function	SGD	ADAM
Number of convolution layers	2	-
First Convolution layer – filter size	13 (1D)	-
First Convolution layer – number of filters	300	-
First Convolution layer – padding size	0	-
First Pooling layer – filter size	5	-
Second Convolution layer – filter size	5 (1D)	-
Second Convolution layer – number of filters	75	-
Second Convolution layer – padding size	0	-
Second Pooling layer – filter size	2	-
Number of Hidden Layers	1	2
Neuron Size in Hidden Layers	300	100
Text Size	100	100

7.6.4. The Results of the BERT Models

Opinion mining experiments with the selected BERT models were performed using the hyperparameters given above.

The BERT models and the experiments can be categorized into three main groups:

1. Base BERT models,

2. BERT models trained for Turkish, and
3. BERT models that were particularly trained for Turkish opinion mining.

The first experiments with BERT models were carried out using the base models; the results are in Table 8. Then, BERT models with multi-language support, including Turkish, were also tested.

Table 8. The results of BERT-base models

No	BERT Model Name	Accuracy [%]
1	BBU	87.34
2	**BBMU**	**88.65**
3	DBMC	88.51
4	RB	86.02

Secondly, experiments were carried out with models trained for Turkish, giving better results than the base models. The results of the 4 Turkish BERT models used are presented in Table 9.

Table 9. The results of BERT models trained for the Turkish

No	BERT Model Name	Accuracy [%]
5	DBTU	92.68
6	DBTC	91.56
7	**D128U**	**92.96**
8	D128C	88.93
9	LBTC	91.28

Finally, models specifically trained for Turkish opinion mining were used, and the best results in the study were obtained with these models. The results obtained with four models developed for Turkish opinion mining are given in Table 10.

Table 108. The results of BERT models trained for the Turkish Opinion Mining

No	BERT Model Name	Accuracy [%]
10	GTSC	92.12
11	KETSA	94.75
12	STSA	90.99
13	**STSC**	**97.19**

As seen in the table above, the best results in the studies were obtained with BERT models, which are based on the Transformer Algorithm. Among the BERT models, it was observed that the models trained for Turkish gave better results than the others, and the best results were achieved among these models, and the BERT models trained specifically for Turkish opinion mining gave the best results. Thus, when choosing the language models to be used, it is seen that it is essential not only to be suitable for the language but also to choose a model suitable for the study area.

8. Discussion

Our experimental results and discussions about our work are given in this section. Firstly, evaluations will be made of the experimental results we obtained using different algorithms and techniques on the same data set. No preprocessing techniques, such as removal of stop words, stemming, and semantic rooting, were used to accurately compare with other algorithm groups and techniques, except for lowercase conversion. Lowercase conversion was used in the preprocessing phase of all Algorithms because it is a technique that increases classification success in all cases compared to other preprocessing methods (Çağataylı, 2015).

First of all, studies were carried out using traditional machine learning algorithms. It has been observed that the most critical hyperparameter for these Algorithms is the choice of a bag of word size. Therefore, this study first found the optimum word bag size to achieve the best results. It has been observed that as the word bag size increases in SVM-rbf, SVM-l and SVM-s, LR, and KNN Algorithms, these Algorithms give more successful results. In the SVM–poly kernel and Naive Bayes Algorithms, it has been observed that contrary to the above result, the accuracy decreases as the bag of words size increases. It has been observed that the results of Decision Tree and Random Forest Algorithms do not change much with the size of the bag of words, and the results remain within a specific range. Hyperparameter values of the traditional Machine Learning Algorithms (e.g., the maximum number of neighbors to be used for KNN) are also optimized for each Algorithm. As a result of numerous optimizations and experiments, it has been shown that approximately 50% to 90% better results can be achieved with traditional machine Algorithms. The best result was obtained with the RBF kernel of the SVM Algorithm. Again, the linear and sigmoid kernels of the SVM Algorithm gave the second and third-best results. LR Algorithm is the algorithm in which we got the fourth best result by providing results that can be used as an alternative to the abovementioned algorithms.

The results we obtained with traditional Machine Learning Algorithms are highly compatible with the previous results we obtained with different data sets in various fields such as text classification, software error classification, and medical text classification (Köksal, 2020; Köksal, 2021; Köksal, 2022; Köksal, 2022-2). However, although the SVM Algorithm gives the best results, different kernels provide the highest success depending on the data set. In the text classification studies mentioned above, the LR Algorithm gave the best result after the SVM Algorithm. While N-gram techniques are used in traditional Machine Learning Algorithms, the highest value obtained in this group (90.32) was reached with the bigram technique.

In the second part of the study, traditional Machine Learning Algorithms were used again, but word representations were used instead of TF-IDF for Vectorization. FastText was chosen as the word representation library (Bojanowski, 2017). FastText provides better results in many NLP tasks than other word representation libraries (Bojanowski, 2017; Joulin, 2017). However, when using the FastText library, the results fell short of the above techniques for this dataset. The main reason is that word representation libraries cannot produce vectors for misspelled words or different vectors. For example, when a comment that should be written as 'super' is deliberately misspelled as 'superr', 'superrr', or 'super', the FastText library produces different vectors for these words, but the classification success decreases.

Turkish opinion mining experiments were conducted with Deep Learning Algorithms in the next stage. In the DNN and CNN algorithms, vectorization is carried out with both TF-IDF and the FastText library. Our previous work (Köksal, 2022-2) stated that the best results were obtained using CNN and the FastText library together.

In the final stage of the study, pre-trained language models based on Transformer Algorithms were used. Experiments were completed using these language models, first the base models, then the models developed for Turkish, and at the last stage, the models developed for Turkish opinion mining. Since the base models were designed only for the English language, they did not yield successful results in Turkish opinion mining as expected. Even in base models trained for multiple languages, including Turkish, only 88.65% was reached. Compared to the base models, the models trained for Turkish were seen to be more successful, and a value of 92.96% was reached in the dbmdz/bert-base-turkish-128k-uncased model, which was trained with extensive data. In the latest experiments, models specific to Turkish opinion mining were used. It has been seen that these models give much more successful results than all other algorithms and techniques. When the most successful model, savasy/bert-turkish-sentiment-cased, was used, the best classification value in the literature in this data set was obtained with a value of 97.19%.

Figure 4 shows the best results we obtained in this study. Finally, the best results we obtained in the experiments were compared with those of previous studies in the literature that had done Turkish opinion mining using the same data set given in Figure 5. It was seen that the best results were obtained in the opinion mining studies carried out on this data set in the literature.

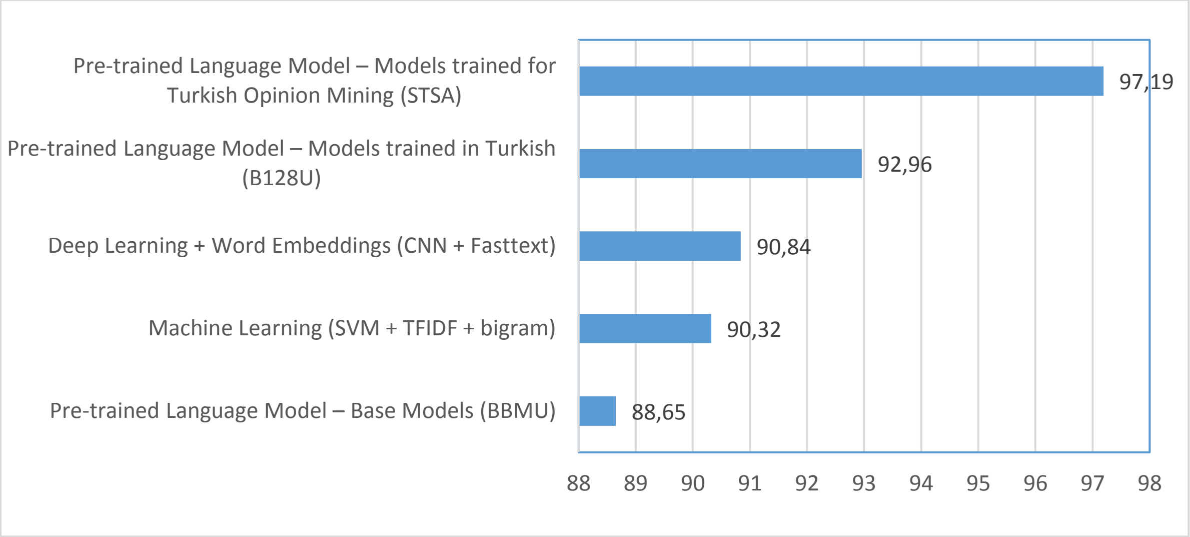

Figure 4. The comparison of the results of algorithm types used

In light of the results given above, the answers to the research results we mentioned above have been clarified: It has been observed that the most successful results are obtained with pre-trained language models based on Transformer Algorithms (RQ.1). Although using word representations for the data set used does not improve the results in traditional Machine Learning Algorithms, it has been observed that more successful classifications can be made when used with Deep Learning Algorithms (RQ.2 and RQ.3).

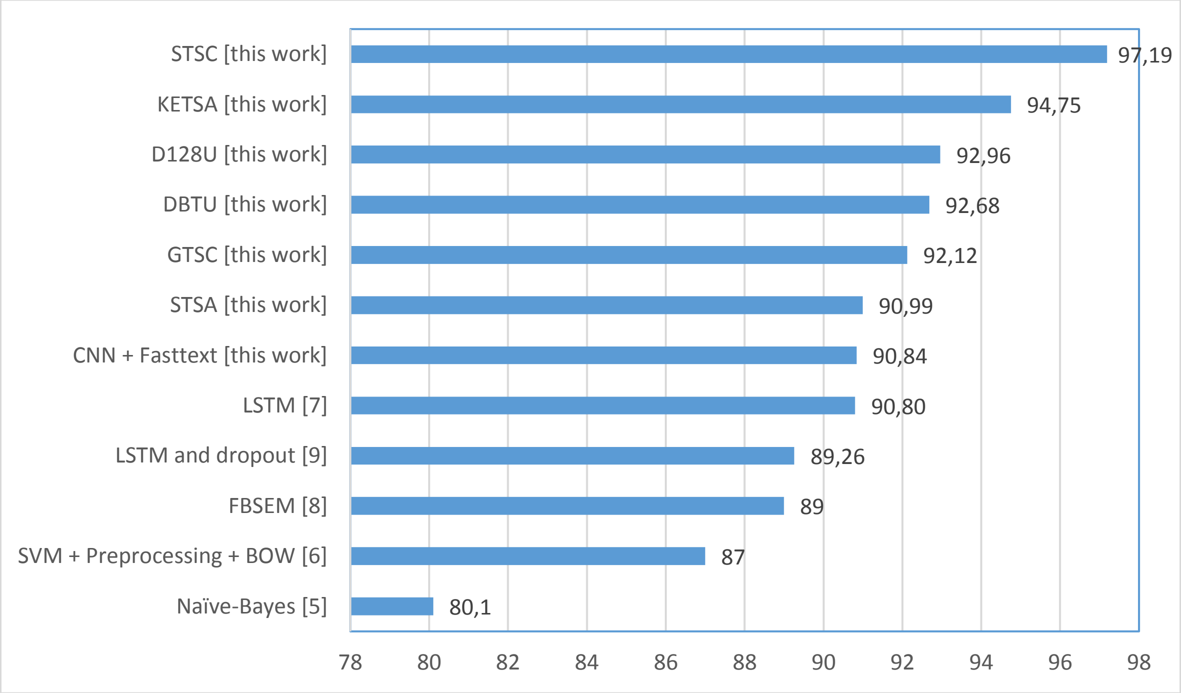

Figure 5. The comparison of experimental results with the results in the literature

It has been observed that pre-trained language models based on Transformer Algorithms can be used in Turkish opinion mining in three different ways: (i) Since the base models do not support the Turkish language, they could not exceed the results of traditional Machine Learning and Deep Learning Algorithms in Turkish opinion mining. The same situation applies to multi-language models where Turkish is included. (ii) It has been observed that when models developed for Turkish are used, more successful results are obtained compared to the base models. (iii) The best results were obtained using models developed especially for Turkish opinion mining. Among the models developed for Turkish opinion mining, it was observed that all four models used in the experiments exceeded all previous results obtained on the same data set in the literature (RQ.3).

9. Conclusions

This study contributes to opinion mining by presenting a novel methodology tailored for less commonly researched languages, addressing their unique challenges. Leveraging a diverse range of artificial intelligence algorithms, including traditional machine learning, deep learning-based approaches, and pre-trained transformer models, our research conducted extensive experiments using an open-source Turkish opinion-mining dataset. We significantly improved classification performance and accuracy scores by meticulously optimizing model parameters and comparing various techniques, such as word embeddings and the traditional bag-of-words method. Our findings outperform existing methodologies in the literature and provide valuable insights for future research in opinion mining, particularly in multilingual contexts.

References

Rumelli M, Akkuş D, Kart O, Işık Z. “Sentiment Analysis in Turkish Text with Machine Learning Algorithms”. Innovations in Intelligent Systems & Applications Conference, ASYU 2019, 2019.

Dehkharghani R, Saygın Y, Yanıkoğlu B, Oflazer K. “SentiTurkNet: a Turkish polarity lexicon for sentiment analysis”. Language Resources & Evaluation, vol. 50, no. 3, pp. 667–685, Sep. 2016.

Çiftçi B, Apaydın MS. “A Deep Learning Approach to Sentiment Analysis in Turkish”. International Conference on Artificial Intelligence & Data Processing, IDAP 2018, 2019.

Açıkalın UU, Bardak B, Kutlu M. “Turkish Sentiment Analysis Using BERT”. 28th Signal Processing & Communications Applications Conference, SIU 2020 - Proceedings, 2020.

Demirtaş E, Pechenizkiy M. “Cross-lingual polarity detection with machine translation”. 2nd International Workshop on Issues of Sentiment Discovery & Opinion Mining, WISDOM 2013 - Held in Conjunction with SIGKDD 2013, 2013.

Gözükara F, Özel SA. “An Experimental Investigation of Document Vector Computation Methods for Sentiment Analysis of Turkish & English Reviews”. Çukurova University, Journal of Engineering and Architecture Faculty, Nov. 2016.

Kurt F, Kısa D, Karagöz P. “Investigating the Effect of Segmentation Methods on Neural Model based Sentiment Analysis on Informal Short Texts in Turkish”. ArXiv, Feb. 2019.

Görmez Y, Işık YE, Temiz M, Aydın Z. “FBSEM: A Novel Feature-Based Stacked Ensemble Method for Sentiment Analysis”. International Journal of Information Technologies, vol. 12, no. 6, pp. 11–22, Dec. 2020.

Yıldırım S. “Comparing Deep Neural Networks to Traditional Models for Sentiment Analysis in Turkish Language”. Deep Learning-based Approaches for Sentiment Analysis, pp. 311–319, 2020.

Işık M, Dağ H. “The impact of text preprocessing on the prediction of review ratings”. Turkish Journal of Electrical Engineering & Computer Science, vol. 28, no. 3, pp. 1405–1421, May 2020.

Gomes LAF, Torres RS, Côrtes ML. “Bug report severity level prediction in open-source software: A survey and research opportunities”. Information and Software Technology, vol. 115. Elsevier B.V., pp. 58–78, 01-Nov-2019.

Alpaydın E. Machine Learning: The New AI. USA, The MIT Press, 2016.

Köksal Ö. “Tuning the Turkish Text Classification Process Using Supervised Machine Learning-based Algorithms”. International Conference on INnovations in Intelligent SysTems and Applications, pp. 1–7, 2020.

Köksal Ö. “Enhancing Turkish sentiment analysis using pre-trained language models”. 29th IEEE Conference on Signal Processing & Communication, 2021.

Köksal Ö, Tekinerdoğan B. “Automated Classification of Unstructured Bilingual Software Bug Reports: An Industrial Case Study Research”. Applied Science, vol. 12, no. 1, 2022.

McMahan RD, Natural language processing with PyTorch: build intelligent language applications using deep learning, USA, O’Reilly, 2019.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L. “Attention Is All You Need”. Neural Information Processing Systems, vol. 2017-Decem, pp. 5999–6009, 2017.

Devlin J, Chang MW, Lee K, Toutanova K. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, 2019.

Köksal Ö, Yılmaz EH. “Improving automated Turkish text classification with learning-based algorithms”. Concurrency & Computation: Practice & Experience, p. e6874, Feb. 2022.

Ambalavanan AK, Devarakonda MV, “Using the contextual language model BERT for multi-criteria classification of scientific articles”. Journal of Biomedical Informatics, vol. 112, Dec. 2020.

Clark K, Luong MT, Le QV, Manning CD. “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”, ArXiv, vol. abs/2003.1, 2020.

OMG, “BPMN Specification - Business Process Model and Notation”, http://www.bpmn.org. (19.02.2024).

Chinosi M, Trombetta A. “BPMN: An introduction to the standard”, Computer Standards & Interfaces, vol. 34, no. 1, pp. 124–134, Jan. 2012.

Mikolov T, Chen K, Corrado G, Dean J. “Efficient estimation of word representations in vector space”. International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, 2013.

Le QV, Mikolov T. “Distributed Representations of Sentences and Documents”. 31st International Conference of Machine Learning. ICML 2014, vol. 4, pp. 2931–2939, May 2014.

Pennington J, Socher R, Manning C. “GloVe: Global Vectors for Word Representation", Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543, 2014.

Bojanowski P, Grave E, Joulin A, Mikolov T. “Enriching Word Vectors with Sub Word Information”. Transactions of the Association for Computer Linguistics., vol. 5, pp. 135–146, Dec 2017.

Çağataylı M, Celebi E. “The effect of stemming and stop-word-removal on automatic text classification in Turkish language”. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9489, pp. 168–176, 2015.

Joulin A, Grave E, Bojanowski P, Mikolov T. “Bag of Tricks for Efficient Text Classification”. 15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 2, Short Papers, pp. 427–431, 2017.

Köksal Ö, O. Akgül, “A Comparative Text Classification Study with Deep Learning-Based Algorithms”. 9th International Conference on Electrical and Electronics Engineering, 2022, pp. 387–39, 2022.