Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
Afyon Kocatepe University – Journal of Science and Engineering
https://dergipark.org.tr/tr/pub/akufemubid

Araştırma Makalesi / Research Article
DOI: https://doi.org/10.35414/akufemubid.1377465
AKÜ FEMÜBİD **24** (2024) 035101 (602-619)
AKU J. Sci. Eng. **24** (2024) 035101 (602-619)

Stance Detection on Short Turkish Text: A Case Study of Russia-Ukraine War

***Makale Bilgisi / Article Info**
Alındı/Received: 17.10.2023
Kabul/Accepted: 08.05.2024
Yayımlandı/Published: 27.06.2024

Türkçe Kısa Metinlerde Duruş Tespiti: Rusya-Ukrayna Savaşı Örneği

Serdar ARSLAN * , **Eray FIRAT**

Çankaya Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye

Abstract
In recent years, social media has emerged as a crucial source of information for gauging public sentiment on a variety of topics. As a result, the need for automated data extraction from these platforms has grown. Stance detection, a subtask in natural language processing, plays a pivotal role in this process by automatically determining users' opinions regarding specific subjects, events, or individuals. To address this, we developed a labeled Turkish dataset focused on determining users' stances on the Russia-Ukraine War using social media content. The dataset, comprising 8215 tweets from Twitter, was meticulously cleaned and annotated for two key targets: Russia and Ukraine. We evaluated several machine learning methods, including Support Vector Machines, Random Forest, k-Nearest Neighbor, XGBoost, Long-Short Term Memory (LSTM), and Gated Recurrent Unit (GRU), with word embeddings from GloVe and FastText. Additionally, we incorporated a transformer-based approach for stance detection. Given the dataset's imbalance between targets, we applied undersampling and oversampling techniques alongside these algorithms. Our experiment results indicate that BERT-based models outperformed all other methods, with LSTM and GRU producing similarly strong outcomes. The newly established Turkish corpus stands as a valuable resource in this field, with potential for future use in conjunction with transformer-based approaches. In summary, this study advances the field of stance detection research in the context of Turkish text.

Keywords *BERT; Deep Learning; NLP; Stance Detection*
.

Öz
Son yıllarda sosyal medya, çeşitli konulardaki halkın görüşlerini anlamak için önemli bir bilgi kaynağı haline gelmiştir. Bu nedenle, bu verilerden otomatik bilgi çıkarmak öneminin arttığı bir alan haline gelmiştir. Doğal dil işleme alanının alt görevlerinden biri olan görüş belirleme, otomatik bilgi çıkarma için kritik bir konudur. Duruş tespiti, kullanıcının belirli bir konu, olay veya kişi hakkındaki tutumunu otomatik olarak belirler. Bu çalışmada, Rusya-Ukrayna Savaşı'na yönelik sosyal medya kullanıcılarının tutumlarını belirleme görevine odaklanan Türkçe etiketli bir veri kümesi oluşturulmuş ve bu veri kümesinde çeşitli makine öğrenimi yöntemleri değerlendirilmiştir. Bu çalışma için 8215 tweet Twitter'dan toplandı ve temizlendi. Veri kümesi daha sonra Rusya ve Ukrayna olmak üzere iki hedefle etiketlendi. Stance Detection görevi için GloVe ve FastText kelime gömme ile Support Vector Machines, Random Forest, k-Nearest Neighbor, XGBoost, Long-Short Term Memory (LSTM) ve Gated Recurrent Unit (GRU) modelleri kullanılmıştır. Ayrıca, duruş tespiti için transformer tabanlı bir yaklaşım da kullanılmıştır. Veri kümesinin hedefler arasındaki dengesizliği dikkate alındığında, bu algoritmalarla birlikte örnek azaltma ve örnek artırma yöntemleri de kullanılmıştır. Deney sonuçları, BERT tabanlı modellerin diğer tüm modelleri geride bıraktığını göstermektedir. Bu sonuçların yanı sıra, LSTM ve GRU da BERT tabanlı modelin sonuçlarına oldukça benzer sonuçlar üretmiştir. Yeni oluşturulan Türkçe veritabanı, bu araştırma alanı için değerli bir kaynak olarak kabul edilebilir ve gelecekte transformer tabanlı yaklaşımlarla birlikte kullanma potansiyeline sahiptir. Özetle, bu çalışma, Türkçe metin bağlamında duruş tespiti araştırma alanını ilerletmektedir.

Anahtar Kelimeler *BERT; Derin Öğrenme; Doğal Dil İşleme; Duruş Tespiti.*

1. 1. Introduction

Stance Detection is commonly regarded as a component of sentiment analysis. Its primary objective is to ascertain an individual's stance concerning a specific target, which can be an explicitly mentioned or implied entity, concept, event, idea, opinion, claim, subject, and more (Mohammad et al. 2016). In contrast to sentiment analysis, Stance Detection primarily centers on discerning a person's standpoint or perspective regarding an evaluative subject, whether it entails supporting or opposing the topic. Stance Detection is intricately connected to various other Natural Language Processing (NLP) tasks that address its subproblems (Küçük and Can 2020).

The field of detecting stance in social media is relatively new in the realm of Natural Language Processing, and there is still ongoing exploration of the influence of

*Sorumlu Yazar/Corresponding Author: Serdar Arslan   e-posta/e-mail: sarslan@cankaya.edu.tr

language and social interactions on a user's stance. Stance-taking has a well-established background in sociolinguistics, with the primary focus being the analysis of an author's viewpoint as expressed in their texts. Fundamentally, the objective of stance determination is to uncover the implicit perspective conveyed within an author's text by considering three key factors: linguistic actions, social interactions, and individual identity (ALDayel and Magdy 2021).

Automating the evaluation of stance has been suggested as a potential initial step in assisting human fact-checkers in identifying false claims (Riedel et al. 2018). Consequently, the Fake News Challenge initiative conducted a competition (FNC-1) to promote the development of algorithms designed to automatically analyze the positions taken by a news source on a specific issue (Pomerleau and Rao 2015). This challenge garnered significant attention from the Natural Language Processing (NLP) community, with participation from 50 teams representing both academia and industry. FNC-1 tasked participants with creating a system that assesses the stance toward a headline when provided with a news article title and its content. The potential stance labels included 'agree,' 'disagree,' 'discuss,' or 'unrelated.' However, it's worth noting that many researchers often simplify this to "Favor," "Against," or "None," frequently omitting the "Unrelated" tag.

As the use of social media platforms like Twitter continues to surge in popularity, the demand for proficient stance detection systems has grown significantly. The exploration of stance detection on social media is in its early stages, with uncertainties about the roles language and social interaction play in deducing a user's stance. Stance detection has a rich history in sociolinguistics, focusing on understanding the writer's perspective through their text. Primarily, the goal of stance detection is to deduce the implicit viewpoint from the writer's text, associating the stance with three key factors: linguistic acts, social interactions, and individual identity (ALDayel and Magdy 2021). These systems have become pivotal in comprehending users' viewpoints and sentiments concerning a wide range of topics. In particular, the ability to detect the stance of Turkish tweets holds great potential for gaining valuable insights into the attitudes and opinions of Turkish-speaking users on diverse issues.

The Stance Detection objective can be expressed through the following equation, involving 'T' for text or 'U' for user and 'G' for the stance tag. The core aim of Automatic Stance Detection is to automatically categorize the author's stance towards a predefined target, assigning one of the available stance tags (ALDayel and Magdy 2021).

$$Stance(T, U|G) = \{Favor, Against, None\} \quad (1)$$

Datasets for stance detection have been predominantly available in the literature for English, and to a lesser extent, for languages like Spanish, Italian, Japanese, Arabic, Russian, Chinese, Catalan, English-Hindi, and Turkish. These datasets initially originated from online forums during the early stages of stance detection research and later transitioned to microblogs (Küçük and Can 2020).

Turkish, due to its complex inflectional and derivational structure, presents distinct challenges in various Natural Language Processing (NLP) tasks compared to languages with simpler morphologies. Previous NLP research on Turkish, benefiting from these unique linguistic characteristics, has paved the way for studies involving similar languages (Yıldırım et al. 2014). In contrast, stance detection research for Turkish is still in its infancy, with only a limited number of studies and datasets focusing on stance detection within the realm of social media and blog posts.

To bridge this gap, our study introduces a dataset consisting of 8215 tweets that capture users' stances on two aspects related to the Russia-Ukraine War. These tweets have been manually annotated for stance, categorizing them as 'favor' or 'against.' In general, in stance detection, the task is often to classify a given piece of text into one of three categories: "favor," "against," or "neutral" (Küçük and Can 2020). Each category represents the stance or opinion expressed in the text regarding a particular subject or topic. By removing the neutral class, our model simplifies the task and focuses on identifying polarized opinions. This can be suitable in scenarios where the goal is to discern between clear positive and negative sentiments or opinions. This is particularly relevant in our domain where neutral stances don't carry significant importance or where the primary interest lies in understanding and analyzing polarized viewpoints.

For the analysis of stance detection on this newly created Turkish dataset, we employed a variety of machine learning methods, including Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Bidirectional Encoder Representations from Transformers (BERT). A crucial aspect of text mining studies is the effective representation of the text content, and thus, we incorporated word embedding models in all these methods.

This work makes several valuable contributions to the field of stance detection and natural language processing, including:

- Creation of a Turkish-Labeled Dataset: The study introduces a novel and manually annotated Turkish dataset specifically designed for stance detection in the context of the Russia-Ukraine conflict. This dataset, comprising over 8,000 tweets, fills a significant gap in Turkish NLP research, providing a valuable resource for future studies.
- Performance Evaluation of Multiple Algorithms: The research evaluates a wide range of machine learning algorithms, including Support Vector Machines, Random Forest, K-Nearest Neighbor, XGBoost, LSTM, GRU, and BERT, on the newly created Turkish dataset. This comparative analysis sheds light on the strengths and weaknesses of different models for stance detection tasks.
- Exploration of Word Embeddings: By incorporating both fastText and GloVe word embeddings, the study explores the impact of different textual representations on model performance. This offers insights into the choice of word embeddings for stance detection tasks.
- Analysis of Imbalanced Data: Stance detection often deals with imbalanced datasets, where one stance significantly outweighs the other. The research addresses this challenge by applying oversampling and undersampling techniques and evaluating their effects on classifier performance. This provides guidance on handling imbalanced data in stance detection tasks.
- Benchmarking with BERT: The study leverages state-of-the-art language model BERT for stance detection and demonstrates its exceptional performance in comparison to other machine learning algorithms. This underscores the potential of transformer-based approaches for stance detection on social media platforms.
- Contributions to Non-English Languages: The findings from this research have broader implications, especially for the study of stance detection in non-English languages. The insights and methodologies developed in this work can be applied to similar languages with complex morphologies.
- Applications in Monitoring and Analysis: The results of this study can be valuable for researchers, policymakers, and journalists interested in monitoring and analyzing social media discourse, particularly related to the Russia-Ukraine conflict. It provides a data-driven approach to understanding public sentiments and attitudes.

In summary, this work contributes to both the practical and theoretical aspects of stance detection, particularly in non-English languages, and highlights the potential of advanced language models like BERT in addressing this challenging NLP task.Overall, the study contributes to the advancement of stance detection research using Turkish text and provides valuable insights into the performance of different machine learning algorithms on stance detection tasks.

Section 2 reviews recent studies related to the stance detection task. The proposed work for Turkish tweets is described in Section 3. The experimental results of the algorithms are compared and discussed in Section 4. Our concluding remarks and possible future work directions are presented in the last section.

2. Related Work

Turkish is characterized by its richness and complexity in terms of morphology, which leads to a distinct set of challenges in various NLP tasks when compared to languages with simpler morphological structures. Consequently, prior NLP research on Turkish has played a pioneering role in the exploration of similar languages that share these morphological complexities (Yıldırım et al. 2014). In contrast, stance detection studies for Turkish are currently an active and open research field, even though there have been only a limited number of studies specifically addressing stance detection within the context of social media and blog posts.

In (Küçük 2017), the author introduced a labeled Turkish stance dataset. This dataset focused on the popular sports clubs Galatasaray and Fenerbahçe in Turkey. Specifically, for these two targets, a dataset was created by collecting tweets, and it included the "Favor" and "Against" stance tags. The dataset was meticulously annotated by a single annotator, resulting in 700 target-tweet pairs for each target. Within these pairs, there were 175 tweets labeled as "Favor" and 175 tweets tagged as "Against." For feature extraction, the Support Vector Machine method was employed with a 10-fold cross-validation procedure, utilizing unigram and hashtag information. Remarkably, this approach demonstrated a high level of performance when compared to similar studies available in the literature.

In (Küçük and Can 2018), authors introduced a new version of the dataset presented in (Küçük 2017), which they labeled as "Version 1." This dataset was reannotated by a different individual. Subsequently, "Version 2" was created, comprising a total of 686 target-tweet pairs, by including tweets that were consistently tagged by two annotators with the same stance label. To further expand the dataset, an additional 400 tweets were collected. Of these, 379 tweets had matching stance tags as assigned by two different annotators, resulting in the creation of "Version 3," which contained a total of 1065 target-tweet pairs.

In their analysis, the Support Vector Machine method was applied with a 10-fold cross-validation technique, utilizing features derived from three different versions, including unigrams, bigrams, hashtags, links, emoticons (e.g., "¡3," ":("), and entity names. Notably, the use of unigram and hashtag features was consistent across all three versions. However, the incorporation of links and emoticons did not significantly contribute to the model's performance. Furthermore, the inclusion of entity names, encompassing person, place, and organization names, was carried out manually using the Named Entity Recognition (NER) Tool. It was observed that the incorporation of entity names did not lead to a discernible improvement in the model's performance.

In (Polat, Güler Bayazıt, and Yildiz 2021), the authors aimed to create a dataset for stance detection in the Turkish language. The dataset was compiled from the well-known Turkish blog site "eksisozluk," which allows unrestricted word usage. This dataset encompasses various topics such as "E-Book," "Working from home," "Mask," "E-Cigarette," "Vaccine," and "Vegan." It consists of 5031 blog posts, distributed unevenly across these topics.

Various text representation methods were employed, including Bag of Words, Term Frequency – Reverse Document Frequency, and Word embedding. The authors conducted an analysis of stance detection results using a range of machine learning methods, such as Naive Bayes, Support Vector Machine, AdaBoost, XGBoost, Random Forest, and Convolutional Neural Networks. Performance evaluation was based on the Matthews Correlation Coefficient. The study found that the most favorable results were achieved with the XGBoost and Convolutional Neural Network methods.

Moreover, the authors applied the integrated gradients method to the features extracted by the Convolutional Neural Network model. This allowed them to analyze the contribution of these features to the prediction performance, providing valuable insights into the stance detection process.

In (Küçük and Arıcı 2022),the authors developed a Turkish dataset sourced from Twitter, focusing on the topic of COVID-19 Vaccination. The dataset was subjected to analysis for both sentiment analysis and stance detection. To construct this dataset, data collection occurred on two separate dates. For the first part (Part-1), 300 tweets were gathered on December 18, 2020, while for the second part (Part-2), an additional 300 tweets were collected on July 18, 2021. The dataset underwent annotation by a single native Turkish annotator for both sentiment and stance classes, ensuring consistent labeling for analysis. The target of the stance detection task in this study is COVID-19 Vaccination. Following the annotation process, the Part-1 dataset was found to have 122 tweets annotated as "Favor," 123 as "Against," and 55 as "None." In the Part-2 dataset, there were 137 tweets annotated as "Favor," 122 as "Against," and 41 as "None." To carry out training and testing, SVM and Random Forest methods were employed, and a 10-fold cross-validation approach was used during the evaluation. This process resulted in a limited number of tweets available for training in each fold, which contributed to relatively lower performance rates. The feature set for stance detection included unigrams, hashtag usage, and emoticon usage. According to the results, the SVM learner outperformed the Random Forest learner, with SVM learners achieving similar performance rates on both the first and second parts of the dataset.

In (Glandt et al. 2021) authors used COVID-19 tweets to detect stance for four different targets by comparing state-of-art methods including BERT, GRU, BiLSTM, CNN and target-specific attention networks. They collected more than thirty thousand (30000) tweets and created an annotated dataset. Then they have established baselines using several supervised learning models.

The study in (Nababan, Mahendra, and Budi 2021) investigates the public's stance on the Job Creation Bill in Indonesia, which has sparked debates. Using keywords related to the bill, nearly 10,000 tweets were collected from Twitter and annotated with stance labels. The study employs Naïve Bayes, Support Vector Machine, and Logistic Regression classifiers with both unigram and bigram features. The highest performance was achieved by the Logistic Regression classifier using unigram features, obtaining a micro F-1 score of 71.8% in the experiments.

The research in (Samih and Darwish 2021) focuses on user stance detection, determining a user's position on a given target, such as a topic or claim. Existing unsupervised classification methods have demonstrated high accuracy (>98%) for vocal Twitter users with numerous tweets on a target but struggle with less vocal users having only a few tweets. The paper presents two approaches to address this issue. In the first approach, user-level stance detection is enhanced by utilizing contextualized embeddings to represent tweets, capturing latent meanings in context. The second approach involves expanding a user's tweets using their Twitter timeline tweets, even if not topically relevant. Unsupervised classification is then performed by clustering the user with others in the training set.

The paper presented in (Grimminger and Klinger 2021) explores the manifestation of the intense social media campaigns and mutual accusations during the 2020 US Elections in online communication, particularly among supporters of candidates Biden and Trump. The study combines tasks of hateful/offensive speech detection and stance detection, annotating 3000 Tweets from the campaign period. The annotations include expressions of stances (favorable, against, mixed, neutral), mentions without an opinion, and identification of offensive style. A BERT baseline classifier indicates high-quality detection of supporters (89% F1 for Trump, 91% F1 for Biden), while identifying those against a candidate is more challenging (79% F1 for Trump, 64% F1 for Biden). Detection of hate/offensive speech remains challenging (53% F1).

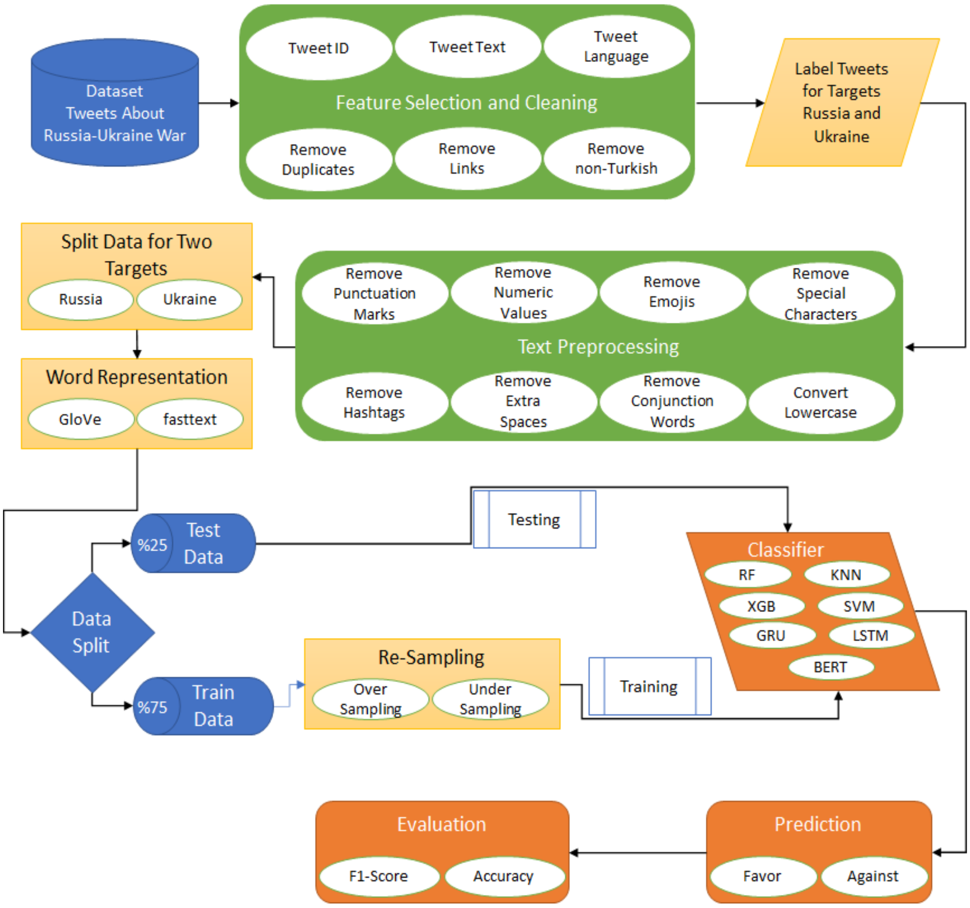

Figure 1. The overall architecture of the proposed system.

The study presented in (Li and Caragea 2019) addresses stance detection, which involves identifying whether the opinion holder is in favor of or against a given target. Recent advancements in stance detection have utilized attention mechanisms or sentiment information to enhance performance. The proposed approach in this paper introduces a multi-task framework that integrates a target-specific attention mechanism while treating sentiment classification as an auxiliary task. Additionally, the model incorporates guidance from a sentiment lexicon and a constructed stance lexicon for the attention layer.

The authors in (Allaway and McKeown 2020) introduce a novel dataset designed for zero-shot stance detection, encompassing a broader spectrum of topics and lexical variations compared to previous datasets. Additionally, a new model is proposed for stance detection, which implicitly captures topic relationships through generalized topic representations. The results demonstrate that this model enhances performance, especially in addressing challenging linguistic phenomena. Stance detection on social media has emerged as a distinct paradigm within opinion mining, especially relevant for various social and political applications where sentiment analysis might fall short. In the literature there are a lot of studies which use supervised learning or unsupervised learning machine learning methodologies and a list of these studies can be found in (ALDayel and Magdy 2021).

3. Proposed Work

In this paper, our primary objective is to assess the accuracy of different models using the dataset we have generated. We conducted experiments involving multiple models and assessed their respective accuracies. The entire process is detailed in Figure 1.

3.1. Dataset collection

Turkey boasts one of the world's largest Twitter user bases, surpassing 16 million users as of June 2022. To perform stance detection in the context of Turkish-Twitter users, we gathered data related to the Russian-Ukrainian War, considering Turkey's diplomatic relationships with both countries.

During the initial days of the war, topics related to "Rusya" (Russia) and "Ukrayna" (Ukraine) became trending topics in Turkey, generating approximately half a million tweets containing these keywords. Subsequently, we collected 13,655 tweets, preprocessed and assigned stances to them, categorizing them as "Favor" or "Against." Tweets that did not exhibit a clear stance were excluded from the labeling process. At the conclusion of this labeling process, we labeled 3,264 tweets related to Ukraine and 4,951 tweets related to Russia.

In adherence to our commitment to transparency and scientific integrity, we are open to sharing the dataset upon a reasonable request, ensuring that appropriate measures are in place to uphold data privacy and confidentiality.

Table 1. Sample tweets.

Tweet	Target	Stance
İşgalci Rusya hesap vereceksin! (EN) (Invading Russia will give account!)	Russia	Against
Rusya'nın operasyonu işgal değil NATO tehdidine karşı savunmadır. (EN) (Russia's operation is not an invasion, but a defense against the NATO threat.)	Russia	Favor
Komedyenden devlet başkanı seçersen, böyle komedi gibi devlet yönetimiyle karşılaşırsın. (EN) (If you choose a comedian for the head of state, you will encounter a state administration like comedy.)	Ukraine	Against
Savaşın her türlüsüne hayır. Ancak dünya burda iki yüzlülüğünü gösterdi, Rusya'ya tepki gösterenler bu yavruların anasını babasını yetim bırakan devletlerdir, Tüm dünya ¸suan Ukrayna için kenetlendik ¸cok güzel, inşallah birgün bu ¸cocuklar için de bütün dünya kenetleniriz... (EN) (No to any kind of war. However, the world showed their hypocrisy here, those who reacted to Russia are the states that orphaned the parents of these puppies, The whole world is now united for Ukraine, it's very nice, I hope one day we will unite for these children as well...)	Ukraine	Favor

Table 2. Word count statistics.

Target	Word Count		
	Min	Max	Average
Russia	2	46	20.8
Ukraine	2	46	21.3

The dataset comprises tweets in which the authors unequivocally conveyed their stance. For each target, each tweet was labeled as either "Favor" or "Against" by four university graduates. Sample target-tweet pairs from the Turkish Stance Labeled Data Set are displayed in Table 1. There were no character limitations imposed on the collected tweets. Table 2 provides insights into the word count statistics for the tweets, including the word count of the shortest tweet, the average word count across tweets, and the word count of the longest tweet for each target. Notably, the generated dataset exhibits an unbalanced class distribution for the targets, as illustrated in Figure 2, depicting the distribution of label numbers for each target.

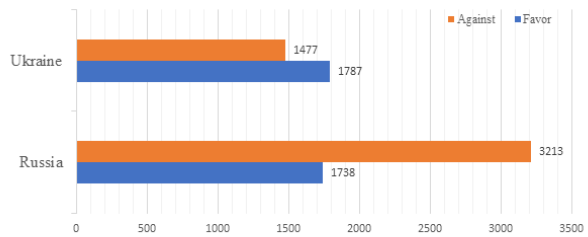

Figure 2. Number of Tweets.

3.2. Data preprocessing

In the classification process, data preparation and cleaning are essential steps to ensure the construction of word representation learning models that work effectively with noise-free and unproblematic text. We executed the preprocessing in two primary stages, both before and after labeling.

During the "before labeling" stage, we retained only Turkish tweets, eliminated duplicates, and removed hyperlinks. In the "after labeling" stage, we further refined the text by removing punctuation marks, numeric values, emojis, special characters, hashtags, extra spaces, and conjunction words. Additionally, all text was converted to lowercase.

It's worth noting that while stemming is commonly recommended for preprocessing in Turkish NLP tasks (Tunali and Bilgin 2012), it is not advised for embeddings. Word embeddings are dense vector representations of words designed to capture semantic and syntactic relationships between words. They are intended to work with the full forms of words rather than their stemmed versions (Jurafsky and Martin 2014).

3.3. Classifiers

In our stance detection process, we employed a variety of algorithms, including K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Machines (SVM), XGBoost, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and BERT. However, these algorithms do not work directly with text data. To make the text data usable with classification algorithms, it must be transformed into numerical representations. This transformation was achieved by converting the text data from tweets into numerical vectors through the utilization of word embedding techniques.

3.3.1. Embedding Layer

Word embedding is a technique commonly employed to represent words as word vectors based on their diverse contextual meanings within sentences. Word vectors capture information from extensive text datasets, allowing words to be efficiently represented as continuous numerical vectors. These word vector models typically assign each word in a dictionary to a specific vector in a mathematical space.

GloVe (Pennington, Socher, and Manning 2014) is another word representation method that takes its name from the initials of the words “Global Vectors for Word Representation”. It is the most used method after Word2Vec(Mikolov et al. 2013) in natural language processing. It was developed by Pennington et al at Stanford University. GloVe is a method that trains on global wordto-word counts, thus allowing statistics to be used more effectively. The GloVe model produces a word space model with 75% accuracy on the analog dataset.

fastText (Bojanowski et al. 2017) is developed by Facebook in 2016. Instead of giving individual words as input to the neural network, it splits words into ”n-grams” based on several letters. For example, for the word “fast”, trigram is “fas” and “ast”. In the N-gram expression, n represents the degree of repetition. In other words, the n expression here provides that we will divide by how many times. It allows us to understand how much of a word or letter. FastText’s word vector is the sum of all these n-gram vectors. After the training is complete, we will have the word vectors for all the n-grams given in the training set.

While GloVe treats each word as the smallest unit to train on, fastText uses ngram characters as the smallest unit. The main advantage of using fastText is that it generates better word embeddings for rare words, or even words not seen (Out of Vocabulary) during training because the n-gram character vectors are shared with other words. GloVe.

3.3.2 Machine Learning Algorithms

K-Nearest Neighbor (KNN) is a supervised machine learning algorithm that can be used for text classification. The algorithm is a non-parametric method that relies on the similarity between the features of the training data and the new input data to make a classification decision. In KNN, the user chooses the number of neighbors (k) to consider when making a prediction for a new observation (Cover and Hart 1967). The algorithm then calculates the distance between the new observation and all training observations and selects the k nearest neighbors based on this distance measure. The most common distance measure used in KNN is the Euclidean distance, but other measures such as Manhattan distance and cosine distance can also be used.

Support Vector Machines (SVM) is a supervised machine learning algorithm that can be used for classification, regression, and outlier detection tasks. The basic idea behind SVM is to find the hyperplane that maximizes the margin between the closest data points from different classes, such that the classification error is minimized (Cortes and Vapnik 1995).

Random Forest (RF) is a machine learning algorithm that uses an ensemble of decision trees to make predictions (Breiman 2001). The algorithm randomly selects a subset of features and a subset of the data to create multiple decision trees. Each tree is trained on a different set of data and features, and the final prediction is based on the majority vote of all the trees.

XGBoost is a popular machine learning algorithm that uses a gradient boosting framework to make predictions (Chen and Guestrin 2016). The algorithm iteratively trains a series of decision trees to improve the accuracy of the predictions. Each tree is trained to correct the errors of the previous tree, and the final prediction is a weighted sum of the predictions from all the trees.

3.3.3. Deep Learning Algorithms

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is designed to overcome the vanishing gradient problem in traditional RNNs (Hochreiter and Schmidhuber 1997). The vanishing gradient problem occurs when the gradient of the error function with respect to the weights of the network becomes very small, which makes it difficult to update the weights and learn long-term dependencies. One of the main advantages of LSTM is its ability to handle long-term dependencies and sequential data (Hochreiter and Schmidhuber 1997). The architecture of LSTM allows it to selectively remember or forget information from previous time steps, which is especially useful for tasks such as speech recognition, language translation, and sentiment analysis.

GRU is a type of recurrent neural network (RNN) and simpler alternative to the LSTM architecture (Cho et al. 2014). Like LSTM, GRU is designed to overcome the vanishing gradient problem in traditional RNNs by allowing the network to selectively forget or remember information from previous time steps. The key difference between GRU and LSTM is the number of gates. While LSTM has three gates (input, output, and forget), GRU has only two gates (update and reset). The update gate controls the amount of new information that is added to the current hidden state, while the reset gate determines the amount of previous information that is discarded.

3.3.4. Transformers

Transformers represent a neural network architecture used for various natural language processing tasks. They are built on the self-attention mechanism (Vaswani et al. 2017), which enables the network to selectively focus on different segments of the input sequence while processing. The central concept in transformers involves calculating an attention weight matrix that indicates the relative importance of each input token in relation to all other tokens. This matrix is then used to weigh the input embeddings and generate the output sequence.

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2019) is a pre-trained transformer-based language model designed for a wide range of natural language processing tasks. It is bidirectional, meaning it considers both the left and right context of each token during training. BERT has demonstrated state-of-the-art performance on various benchmarks and has become a popular choice for NLP tasks. There is also a Turkish BERT model called BerTurk, which is pre-trained on datasets like the Oscar Corpus, Opus Corpora, and Wikipedia dump. The model comprises 12 transformer layers, and it comes in different versions, including those with 32K and 128K vocabulary sizes, as well as cased and uncased variants.

3.4. Evaluation

In literature,recall (eq.2), precision (eq.3), and accuracy (eq.4) obtained from confusion matrix (Table 3) , are used as common metrics to evaluate the performance of the models. However, the class imbalance has a large effect on measurements and dominant class has negatively affects accuracy. Therefore, f1 score (eq.5), which is the harmonic mean of precision and recall is commonly used to compare results in studies on unbalanced dataset.

Table 3. Confusion matrix.

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (TN)	True Negative (TN)

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

3.5. Experiments

The tweets in our dataset were converted into vector representations using two different word embeddings: fastText and GloVe. We used GloVe embeddings with 200 dimensions and fastText embeddings with 300 dimensions.

Table 4. XGBoost hyperparameters.

Parameter	Value
learning rate	0,1
max depth	7
n_estimators	80
eval_metric	auc

In our experiments, we applied various traditional machine learning algorithms, including K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) to work with these word embeddings. For KNN, we utilized the Euclidean distance metric, commonly preferred in text classification tasks. Default parameters of the scikit-learn library (Pedregosa et al. 2011) were used for the other algorithms.Furthermore, we implemented an XGBoost model using both fastText and GloVe embeddings to transform the tweets into vector representations. The XGBoost model was fine-tuned with pre-optimized hyperparameters, which are detailed in Table 4 of our study.In our study, we developed two LSTM models and two GRU models, each utilizing two different word embeddings: fastText and GloVe. We employed single-layer LSTM and single-layer GRU architectures, both with a 128-hidden layer. The dropout rate was set at 0.5, although it had no significant impact on our results. These models were compiled with the ADAM optimizer, using a learning rate of 0.001 and binary cross-entropy as the loss function. The final output layer contained a single neuron with sigmoid activation. The models were trained with a batch size of 64 and early stopping criteria.

In the latter part of our study, we incorporated the Bert Model. The Bert model consists of two stages: pre-training and fine-tuning. During the pre-training phase, the model learns from unsupervised text data through various pre-training tasks, such as the masked language model and predictions of the next sentence. In the fine-tuning phase, the Bert model is initialized with parameters from the pre-training stage and adjusts these parameters using labeled data from downstream tasks. We applied Grid Search mechanism for parameter selection and the parameters with best results have been used. The architectural design of the Bert model includes an encoder with 768 hidden size, 12 self-attention heads, and 12 transformer blocks. Bert produces sequence representations for input sequences of up to 512 tokens. We utilized the 128K uncased BERTurk model with a learning rate of 5e-5 and a batch size of 32 for training, conducting four epochs.

Table 5. Experiment results.

Classifier	Embeddings	Russia		Ukraine	
		Accuracy	**F1 Score**	**Accuracy**	**F1 Score**
KNN	fastText	0.632	0.639	0.593	0.564
	Glove	0.588	0.597	0.640	0.610
RFC	fastText	0.696	0.639	0.760	0.759
	Glove	0.672	0.585	0.749	0.746
SVM	fastText	0.721	0.686	0.809	0.809
	Glove	0.700	0.645	0.767	0.767
XGBoost	fastText	0.717	0.693	0.782	0.782
	Glove	0.691	0.651	0.771	0.769
LSTM	fastText	0.713	0.712	0.752	0.752
	Glove	0.680	0.687	0.807	0.807
GRU	fastText	0.683	0.684	0.755	0.755
	Glove	0.687	0.685	0.807	0.807
BERT		**0.784**	**0.787**	**0.872**	**0.870**

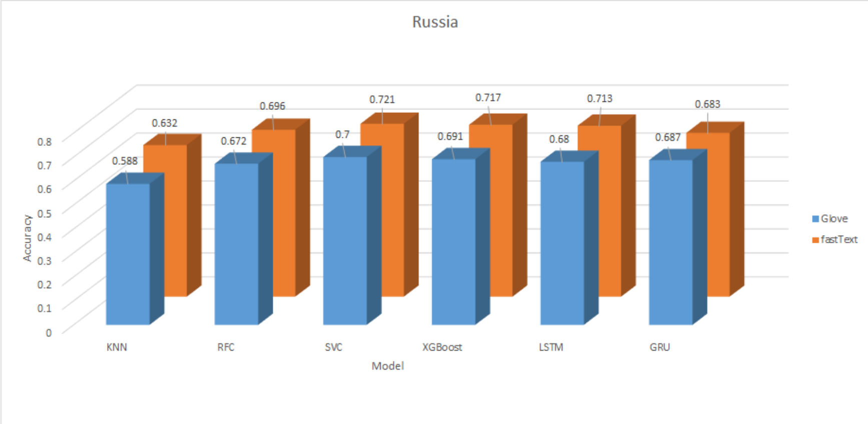

Figure 3. Glove and fastText embeddings accuracy results of models for Russia data set (except BERT).

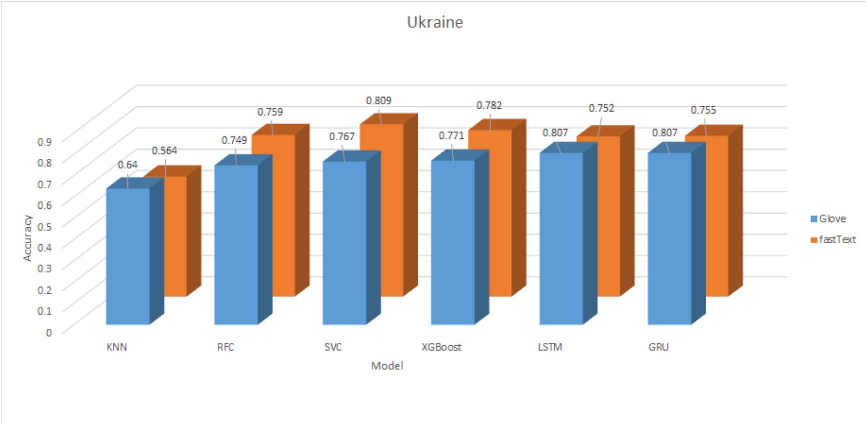

Figure 4. Glove and fastText embeddings accuracy results of models for Ukraine data set (except BERT).

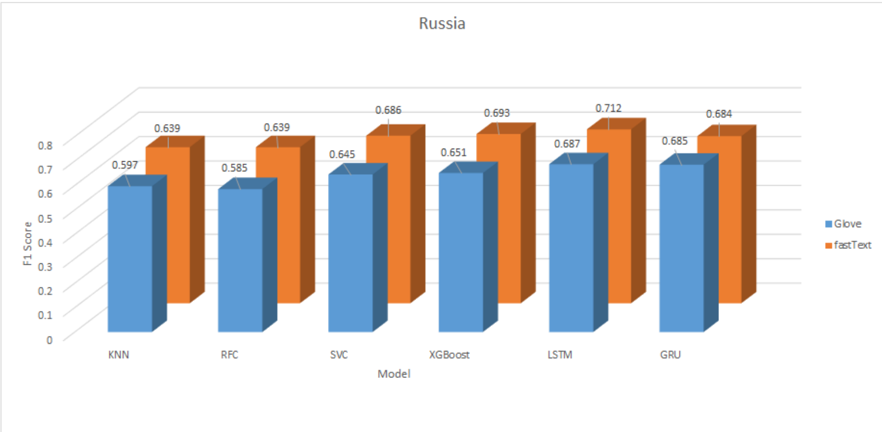

Figure 5. Glove and fastText embeddings F1 score results of models for Russia data set (except BERT).

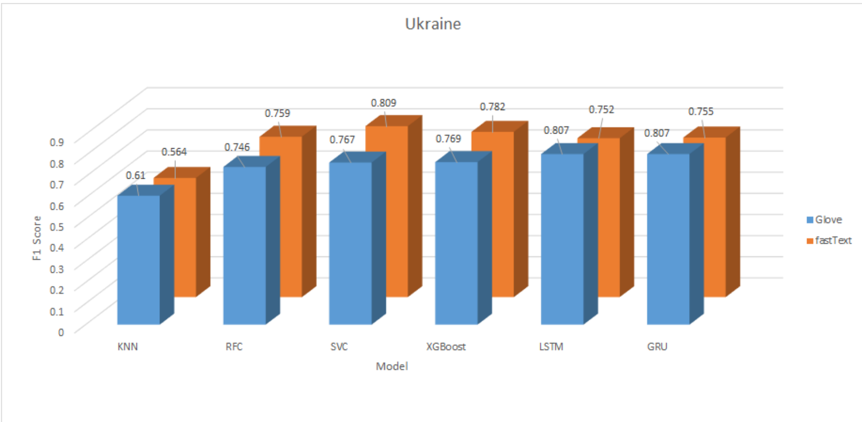

Figure 6. Glove and fastText embeddings F1 score results of models for Ukraine data set (except BERT).

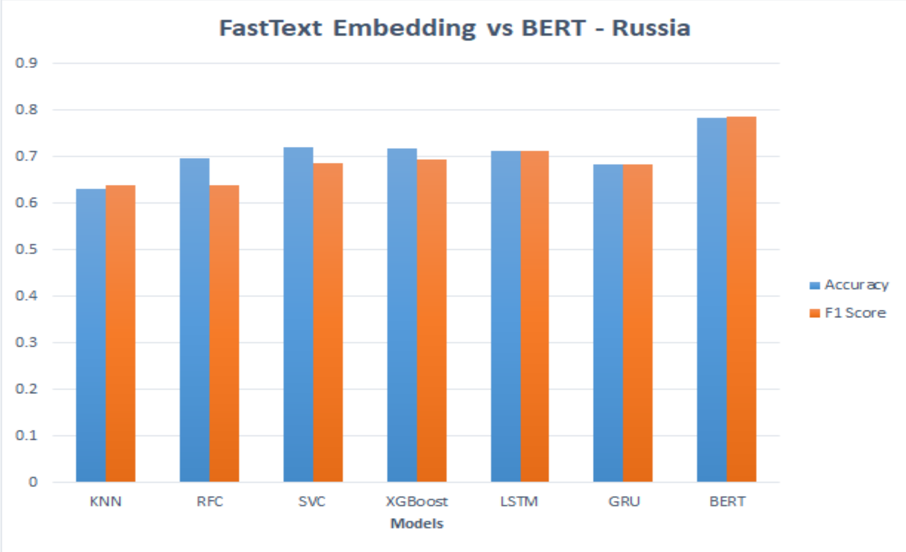

Figure 7. Comparison of models for Russia data set using fastText embedding.

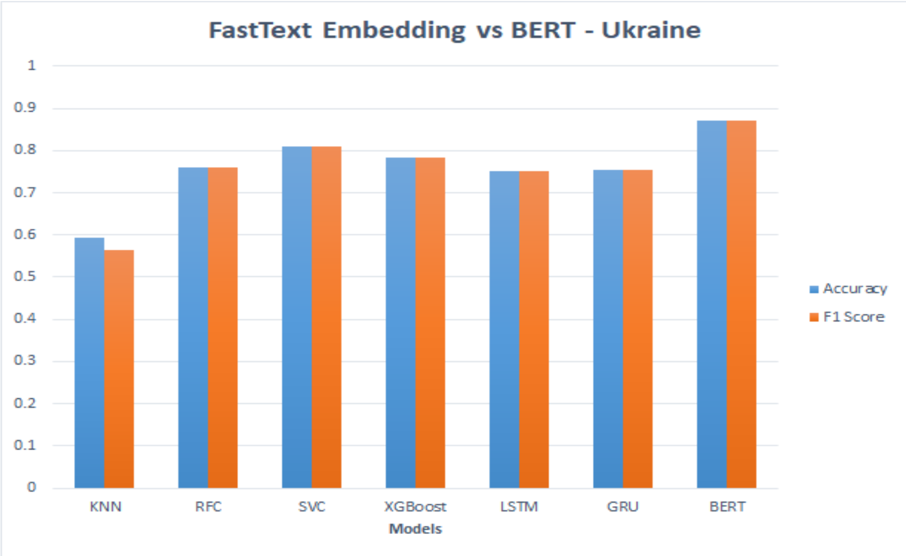

Figure 8. Comparison of models for Ukraine data set using fastText embedding.

The study's results, presented in Table 5, display the accuracy and F1 scores of various classifiers trained on two distinct datasets, one for Russia and another for Ukraine. These classifiers were trained using two different word embedding techniques: fastText and GloVe. The models employed included K-Nearest Neighbors (KNN), Random Forest Classifier (RFC), Support Vector Machine (SVM), XGBoost, Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and BERT. The effects of embedding method on results are demonstrated in Figure 3 – 6. The BERT models are not included in these figures, since we are comparing fastText and Glove embeddings. Figure 7 and Figure 8 show the comparison results of BERT with other models which use fastText embedding. Similarly Figure 9 and Figure 10 show the comparison results of BERT with other models which use Glove embedding.

One notable observation is that the classifiers achieved superior performance on the Ukraine dataset compared to the Russia dataset. This discrepancy could be attributed to the Ukraine dataset's more balanced data distribution during training, which benefits the models. In terms of classification models, Support Vector Machine (SVM) outperformed the other models for both targets. This outcome aligns with prior studies on stance detection, where SVM consistently demonstrated strong performance across various languages and domains. The success of SVM can be attributed to its simplicity, effectiveness in handling high-dimensional data, and resistance to overfitting.

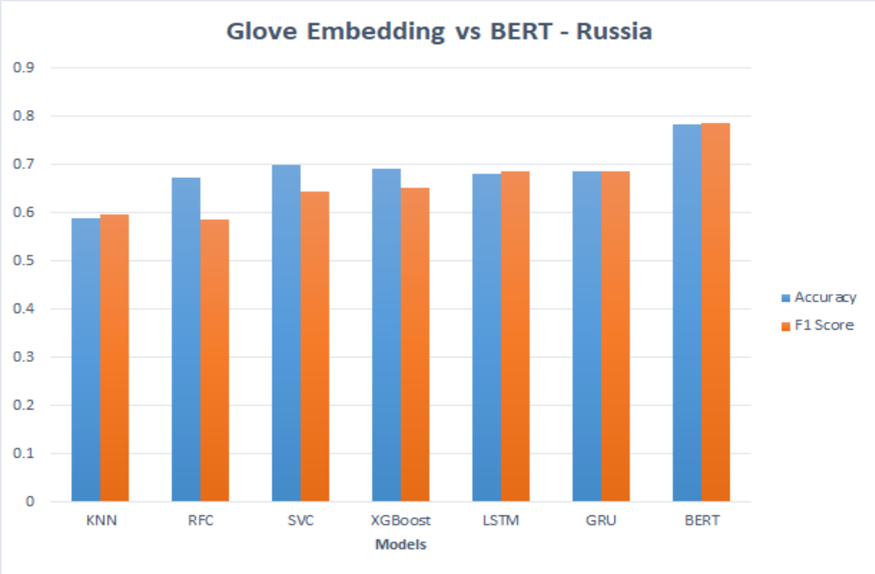

Figure 9. Comparison of models for Russia data set using Glove embedding.

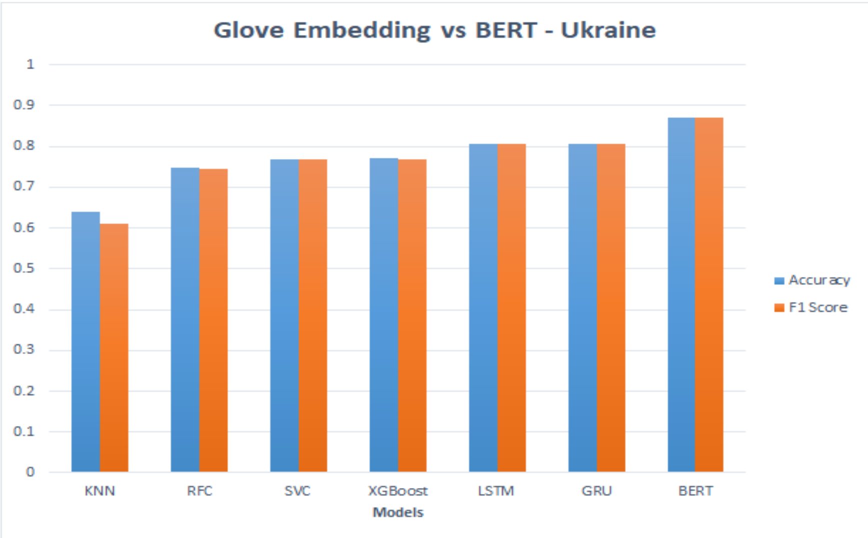

Figure 10. Comparison of models for Ukraine data set using Glove embedding.

Ultimately, the BERT model, a cutting-edge language model grounded in transformer architecture, outperformed all other models by attaining the highest accuracy and F1-score for both targets. This outcome aligns with recent research highlighting BERT's efficacy in a range of natural language processing tasks (Devlin et al. 2019). BERT, as a pre-trained language model, is amenable to fine-tuning for specific applications, including stance detection, and excels in capturing the contextual information within tweets. Due to the dataset's inherent imbalance and the outcomes of our initial experiments, we decided to conduct supplementary experiments involving oversampling and undersampling techniques. We applied the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al. 2002) for oversampling and the NearMiss technique for undersampling. Oversampling seeks to balance the class distribution by generating synthetic samples for the minority class to align with the majority class's sample count. In contrast, undersampling reduces the number of samples in the majority class to match the minority class's count.

We employed the same models as in the initial experiments on both the oversampled and undersampled datasets, evaluating their performance using the same metrics. The results, as presented in Table 6, indicate that the oversampling technique enhanced the performance of most models for both targets, whereas the undersampling technique did not yield significant improvements. Figure 11 and Figure 12 demonstrate the performance comparison of BERT with fastText embedded models using undersampling and oversampling methods. Figure 13 and Figure 14 show the same metrics for Glove embedding.

Table 6. Experiment results with Oversampling and Undersampling.

Classifier	Embeddings	Russia		Ukraine	
		Accuracy	F1 Score	Accuracy	F1 Score
KNN oversampling	fastText	0.573	0.576	0.599	0.581
	Glove	0.557	0.561	0.625	0.606
KNN undersampling	fastText	0.552	0.547	0.602	0.583
	Glove	0.531	0.520	0.641	0.626
RFC oversampling	fastText	0.712	0.681	0.754	0.754
	Glove	0.699	0.657	0.754	0.751
RFC undersampling	fastText	0.662	0.668	0.745	0.746
	Glove	0.655	0.662	0.744	0.744
SVM oversampling	fastText	0.728	0.732	0.805	0.805
	Glove	0.715	0.720	0.775	0.775
SVM undersampling	fastText	0.732	0.737	0.803	0.803
	Glove	0.694	0.700	0.763	0.764
XGBoost oversampling	fastText	0.712	0.699	0.773	0.773
	Glove	0.705	0.690	0.762	0.762
XGBoost undersamplin	fastText	0.679	0.685	0.761	0.762
	Glove	0.661	0.668	0.768	0.769
LSTM oversampling	fastText	0.697	0.699	0.781	0.781
	Glove	0.680	0.687	0.795	0.796
LSTM undersampling	fastText	0.667	0.671	0.766	0.765
	Glove	0.668	0.673	0.790	0.790
GRU oversampling	fastText	0.680	0.681	0.796	0.797
	Glove	0.674	0.680	0.819	0.820
GRU undersampling	fastText	0.655	0.659	0.761	0.761
	Glove	0.686	0.690	0.786	0.787
BERT oversampling		**0.774**	**0.776**	0.813	0.808
BERT undersampling		0.740	0.746	**0.835**	**0.834**

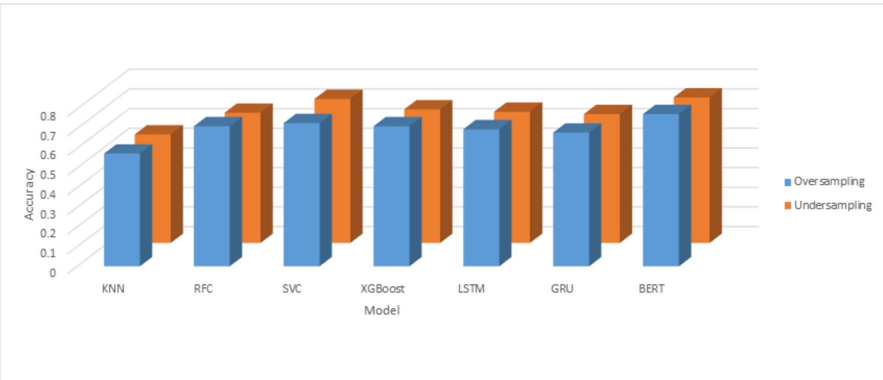

Figure 11. Comparison of models for Russia data set using fastText embedding and oversampling/undersampling.

Figure 12. Comparison of models for Ukraine data set using fastText embedding and oversampling/undersampling.

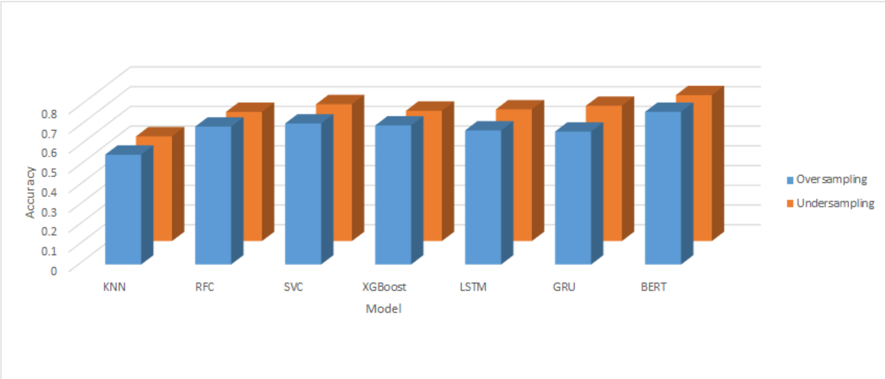

Figure 13. Comparison of models for Russia data set using Glove embedding and oversampling/undersampling.

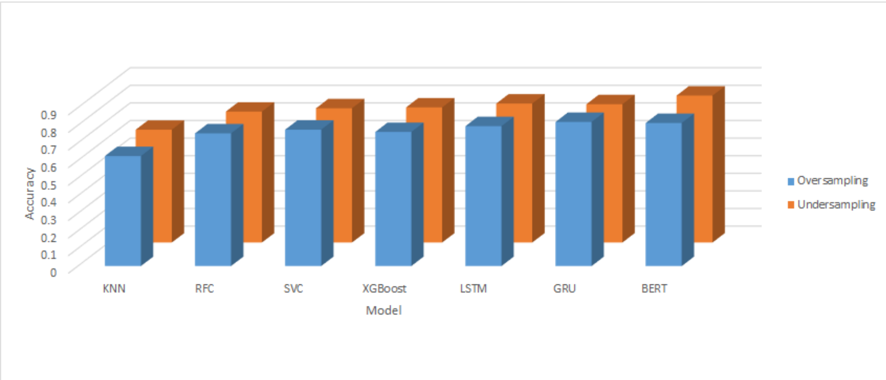

Figure 14. Comparison of models for Ukraine data set using Glove embedding and oversampling/undersampling

When analyzing the impact of oversampling and undersampling on classifier performance, we observed diverse effects. Some models exhibited improvements in accuracy and F1-score, while others experienced a decline in performance due to these techniques. For instance, the K-Nearest Neighbor (KNN) classifier using GloVe embeddings with undersampling achieved higher accuracy and F1-score for Ukraine, while the same classifier using GloVe embeddings with undersampling resulted in lower accuracy and F1-score for Russia. In contrast, the Random Forest classifier with undersampling, utilizing either GloVe or fastText embeddings, consistently exhibited a decrease in accuracy and F1-score for both countries. Conversely, the Support Vector Machine (SVM) classifier with oversampling, using either GloVe or fastText embeddings, consistently demonstrated improvements in accuracy and F1-score for the Russia target. It's essential to recognize that the effectiveness of oversampling and undersampling methods can be influenced by the specific dataset and classifier used. As a result, it is advisable to conduct experiments with both techniques to determine which one yields the best performance for a given task.

4. Discussion

In our study, we aimed to develop a robust stance detection model for newly generated Turkish tweets concerning the Russia-Ukraine conflict. Understanding social media users' perspectives on this matter is of great interest to researchers, policymakers, and journalists. However, this task comes with its challenges, primarily due to the imbalanced dataset where one stance dominates the other. To tackle this challenge, we explored various machine learning algorithms and techniques to create an effective and efficient model.

We conducted experiments with six distinct machine learning algorithms, including support vector machines, random forest, k-nearest neighbor, XGBoost, LSTM, and GRU, utilizing word embeddings from fastText and GloVe. To address the dataset's imbalance, we applied both undersampling and oversampling methods. Our experimental findings revealed that support vector machines with fastText and undersampled data delivered the best performance in detecting the stance of tweets related to Russia, achieving an F1 score of 0.738. In the case of tweets concerning Ukraine, support vector machines with fastText also outperformed other models, boasting an F1 score of 0.809. Notably, we observed that LSTM and GRU exhibited performance very close to support vector machines when detecting the stance of tweets related to Ukraine.

Additionally, we assessed the performance of the recently introduced state-of-the-art BERT model on our dataset. By fine-tuning the pre-trained BERT model on our data, we observed a substantial enhancement in performance when compared to other machine learning algorithms. BERT achieved the highest performance for both targets, achieving an impressive F1 score of 0.787 for Russia and 0.870 for Ukraine, surpassing the performance of other models significantly. These results emphasize the potential of fine-tuned BERT as a promising approach for stance detection on social media platforms. Therefore we can conclude that BERT, being a transformer-based model, has shown remarkable success in understanding contextual information in language, making it particularly effective in tasks like stance detection. BERT excels in capturing contextual information in text, considering the entire context of a sentence rather than relying solely on word embeddings or fixed-size context windows. This is crucial in stance detection task where the meaning may depend on the overall context of the statement.

Moreover, BERT is pre-trained on a large corpus, learning rich representations of language. This pre-training allows it to capture complex linguistic patterns and relationships, making it highly effective in this study without requiring extensive task-specific labeled data.

Additionally, BERT's architecture allows for fine-tuning on specific tasks, making it adaptable to the nuances of stance detection in Turkish tweets. This flexibility is crucial in adjusting the model to the characteristics of the target dataset.

Finally, BERT utilizes attention mechanisms that enable it to focus on relevant parts of the input sequence, effectively capturing dependencies between words. This attention mechanism is particularly advantageous in understanding the nuanced language often present in social media and tweets.

5. Conclusion

In recent times, social media has become an indispensable source for understanding public sentiments across various domains, necessitating automated methods for extracting relevant data from these platforms.

Stance detection, a crucial aspect of natural language processing, assumes a central role in this endeavor by automatically discerning users' viewpoints on specific subjects, events, or personalities. To contribute to this area, we curated a labeled dataset in Turkish, focusing on capturing users' stances pertaining to the Russia-Ukraine War through social media discourse. This meticulously curated dataset comprises 8215 tweets sourced from Twitter, subjected to thorough cleaning and annotation with respect to two focal points: Russia and Ukraine.

Our study involved the evaluation of diverse machine learning methodologies, encompassing Support Vector Machines, Random Forest, k-Nearest Neighbor, XGBoost, alongside recurrent neural network architectures such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), utilizing word embeddings from GloVe and FastText.

Additionally, we explored the effectiveness of a transformer-based approach for stance detection. Given the inherent imbalance within the dataset across different targets, we employed both undersampling and oversampling techniques in conjunction with these algorithms.

Our experimental findings underscored the superior performance of BERT-based models over alternative methods, with LSTM and GRU models also yielding notable results. The establishment of this Turkish corpus signifies a significant contribution to the realm of stance detection research, particularly concerning Turkish text, and holds promise for future investigations, especially when combined with transformer-based methodologies. In essence, this study represents a stride forward in advancing stance detection research within the context of Turkish language analysis.

One limitation of this work is nature of the data set. Stance annotation can be subjective, leading to potential disagreement among annotators. It's essential to assess and address the inter-annotator agreement to understand the reliability of the dataset. Due to this subjectivity of the topic and also manual annotations, imbalanced datasets can pose challenges, especially if one stance is significantly more prevalent than the other. While this study addresses this through oversampling and undersampling, it's crucial to explore additional techniques or data augmentation methods.

Another future work can be generating the model for different data sets in different domains. Our dataset is focused on the Russia-Ukraine conflict. Stance detection models trained on this specific domain may not generalize well to other topics or domains. Creating or using more diverse datasets may enhance the models' generalizability.

Although BERT demonstrates impressive performance, it's essential to consider that the optimal model choice may vary based on factors such as dataset size, computational resources, and task-specific characteristics. In instances where data is limited, simpler models like SVM or traditional machine learning classifiers might yield satisfactory results.

As a result, BERT's exceptional performance in stance detection for Turkish tweets is consistent with its success across diverse NLP tasks. It highlights the capability of transformer-based models to grasp intricate language structures and contextual nuances, particularly in contexts requiring a deep understanding of nuanced opinions and sentiments, such as social media text.

To conclude, this study demonstrates the feasibility and effectiveness of stance detection on newly generated Turkish tweets related to the Russia-Ukraine conflict, even with an imbalanced dataset. Our research sheds light on the effectiveness of various machine learning algorithms and techniques for stance detection tasks, particularly in non-English languages. These findings can provide valuable insights for researchers, policymakers, and journalists who are interested in monitoring and analyzing social media discourse related to the Russia-Ukraine conflict.

Declaration of Ethical Standards

The authors declare that they comply with all ethical standards.

Credit Authorship Contribution Statement

Author-1: Conceptualization, investigation, writing – review and editing and supervision.

Author-2: Conceptualization, investigation, methodology and software, data curation and writing – original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability Statement

Datasets are available on request. The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

6. References

ALDayel, Abeer, and Walid Magdy. 2021. Stance Detection on Social Media: State of the Art and Trends. *Information Processing and Management* **58(4)**:102597. https://www.doi.org/10.1016/j.ipm.2021.102597.

Allaway, Emily, and Kathleen McKeown. 2020. Zero-Shot Stance Detection: A Dataset and Model Using Generalized Topic Representations. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* 8913–31. https://www.doi.org/10.18653/v1/2020.emnlp-main.717.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. "Enriching Word Vectors with Subword Information."

Breiman, Leo. 2001. Random Forests. *Machine Learning* **45(1)**:5–32.

https://www.doi.org/10.1023/A:1010933404324.

Chawla, N. V, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. {SMOTE}: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* **16**:321–57. https://www.doi.org/10.1613/jair.953.

Chen, Tianqi, and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *KDD '16*. New York, NY, USA: Association for Computing Machinery, 785–794.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. BT - Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A Meeting of SIGDAT,. 1724–1734.

Cortes, Corinna, and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning* **20(3)**:273–97.

https://www.doi.org/10.1007/BF00994018.

Cover, T., and P. Hart. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* **13(1)**:21–27. https://www.doi.org/10.1109/TIT.1967.1053964.

Devlin J., Chang MW., Lee K., Toutanova K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, , Minneapolis, Minnesota. Association for Computational Linguistics, **1**, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Glandt K., Khanal S., Li y., Caragea D., Caragea C., 2021. Stance Detection in COVID-19 Tweets. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference* 1596–1611. https://www.doi.org/10.18653/v1/2021.acl-long.127

Grimminger, L., Klinger R., 2021. Hate Towards the Political Opponent: A Twitter Corpus Study of the 2020 US Elections on the Basis of Offensive Speech and Stance Detection. *WASSA 2021 - Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Proceedings of the 11th Workshop* 171–180.

Hochreiter, S., Schmidhuber J., 1997. Long Short-Term Memory. *Neural Comput.* **9(8)**:1735–80. https://www.doi.org/10.1162/neco.1997.9.8.1735.

Jurafsky, D., Martin J., 2014. *Speech and Language Processing*. Vol. 3.

Küçük, D., 2017. “Stance Detection in Turkish Tweets.” *CEUR Workshop Proceedings* **1914**:3–6. https://www.doi.org/10.475/123.

Küçük, D, Can F., 2018. Stance Detection on Tweets: An SVM-Based Approach. https://doi.org/10.48550/arXiv.1803.08910

Küçük, D., Can f.,. 2020. Stance Detection: A Survey. *ACM Comput. Surv.* **53(1)**. https://www.doi.org/10.1145/3369026

Küçük, Doğan, and Nursal Arıcı. 2022. Sentiment Analysis and Stance Detection in Turkish Tweets About COVID-19 Vaccination.371–87.

Küçük D., Arıcı N., 2022, Sentiment Analysis and Stance Detection in Turkish Tweets About COVID-19 Vaccination, In Handbook of Research on Opinion Mining and Text Analytics on Literary Works and Social Media, Arizona: IGI-Global, 371-387.

Li, Y., Caragea C., 2019. Multi-Task Stance Detection with Sentiment and Stance Lexicons. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference,* 6299–6305. https://www.doi.org/10.18653/v1/d19-1657.

Mikolov, T., .Chen K, Corrado G., Dean J., 2013. Efficient Estimation of Word Representations in Vector Space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings,* 1–12.

Mohammad S., Kiritchenko S., Sobhani P., Zhu X., Cherry C., 2016. A Dataset for Detecting Stance in Tweets. *Proceedings of the Tenth International Conference on Language Resources and Evaluation ({LREC}’16)*. Portorož, Slovenia: European Language Resources Association (ELRA), 3945–3952.

Nababan, AH., Mahendra R., Budi I., 2021. Twitter Stance Detection towards Job Creation Bill. *Procedia Computer Science* **197(2021)**:76–81. https://www.doi.org/10.1016/j.procs.2021.12.120.

Pedregosa, F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderolas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **12(85)**, 2825–2830.

Pennington, J., Socher R., Manning C., 2014. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing ({EMNLP})*. Doha, Qatar: Association for Computational Linguistics. 1532–1543

Polat, K.K, Bayazıt N.G., Yildiz O.T., 2021. Türkçe Duruş Tespit Analizi. *European Journal of Science and Technology,* **(23),** 99–107. https://www.doi.org/10.31590/ejosat.851584.

Pomerleau, D., Rao D., 2015. Fake News Challenge. *2015* http://fakenewschallenge.org/. Retrieved (http://www.fakenewschallenge.org/).

Riedel, B., Augenstein I., Spithourakis G.P., Riedel S., 2018. A Simple but Tough-to-Beat Baseline for the Fake News Challenge Stance Detection Task. ArXiv: 1707.03264 https://doi.org/10.48550/arXiv.1707.03264

Samih, Y., .Darwish K., 2021. A Few Topical Tweets Are Enough for Effective User Stance Detection. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference,* 2637–46. https://www.doi.org/10.18653/v1/2021.eacl-main.227.

Tunali, V., Bilgin T.T., 2012. Examining the Impact of Stemming on Clustering Turkish Texts. *2012 International Symposium on Innovations in Intelligent Systems and Applications*, 1–4.

Vaswani, A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I., 2017. "Attention Is All You Need." in *Advances in Neural Information Processing Systems*, **30**., 1-11

Yıldırım, E., Çetin F.S., Eryiğit G., Temel T., 2014. The Impact of NLP on Turkish Sentiment Analysis. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi,* **7(1)**:43–51.