IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 6 November 2023, accepted 23 November 2023, date of publication 28 November 2023,
date of current version 5 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337354

RESEARCH ARTICLE

Analysis of Deep Learning Model Combinations and Tokenization Approaches in Sentiment Classification

ALI ERKAN, (Member, IEEE), AND TUNGA GÜNGÖR
Department of Computer Engineering, Boğaziçi University, Istanbul 34342, Turkey

Corresponding author: Ali Erkan (ali.erkan@boun.edu.tr)

This work was supported by the Turkish Directorate of Strategy and Budget under TAM Project under Grant 2007K12-873.

ABSTRACT Sentiment classification is a natural language processing task to identify opinions expressed in texts such as product or service reviews. In this work, we analyze the effects of different deep-learning model combinations, embedding methods, and tokenization approaches in sentiment classification. We feed non-contextualized (Word2Vec and GloVe) and contextualized (BERT and RoBERTa/XLM-RoBERTa) embeddings and also the output of the pretrained BERT and RoBERTa/XLM-RoBERTa models as input to neural models. We make a comprehensive analysis of eleven different tokenization approaches, including the commonly used subword methods and morphologically motivated segmentations. The experiments are conducted on three English and two Turkish datasets from different domains. The results show that BERT- and RoBERTa-/XLM-RoBERTa-based and contextualized embeddings outperform other neural models. We also observe that using words in raw or preprocessed form, stemming the words, and applying WordPiece tokenizations give the most promising results in the sentiment analysis task. We ensemble the models to find out which tokenization approaches produce better results together.

INDEX TERMS Machine learning, deep neural networks, natural language processing, sentiment classification, word embedding, tokenization, morphology.

I. INTRODUCTION

Sentiment classification is a natural language processing task of determining and understanding a text's emotions, attitudes, and opinions in social media posts, customer reviews, news articles, and the like. It involves analyzing the text's language, context, and tone to identify whether the sentiment expressed is positive, negative, or neutral. The sentiment of a text provides valuable information for various applications. Today, comments posted on the Web are far beyond the scale that people can handle manually. For this reason, it has become essential to automate the sentiment analysis task. In addition to obtaining positive and negative sentiments, sentiment analysis studies are also conducted to extract emotions from texts. For example, Raman et al. [1] analyzed the performance of different algorithms in detecting aggression and hate speech from Twitter and Facebook messages. Similarly, Rana et al. [2] studied sentiment analysis based on customer loyalty by using text mining and machine learning models to extract customer feelings.

The importance of sentiment analysis lies in its ability to extract valuable insights and make informed decisions in various domains. Here are some areas where sentiment analysis is important and the key reasons:

- Customer feedback and reputation management: Sentiment analysis allows businesses to monitor and analyze customer feedback in real time. It helps manage reputation, enhance customer satisfaction, and maintain brand loyalty.
- Market research and competitive analysis: Companies can identify market trends, understand customer preferences, and gain a competitive edge by analyzing sentiment across different demographics, locations, or time periods. It helps in decision making, product development, and marketing strategies.

The associate editor coordinating the review of this manuscript and approving it for publication was Mahdi Zareei.

- Social media monitoring: By tracking mentions and analyzing sentiment, businesses can gauge the effectiveness of their social media campaigns, identify potential crises, and engage with customers in real time. It helps in brand management, crisis response, and social media marketing.
- Financial analysis and stock market prediction: Sentiment analysis is employed in the financial sector to analyze news articles, social media posts, and financial reports to gauge market sentiment.
- Public opinion and policy analysis: Sentiment analysis is used in politics and governance to analyze public sentiments and opinions toward government policies, public figures, or political events.

Although rule-based approaches exist to extract sentiment from a document, the studies showed that machine learning algorithms are superior in obtaining more accurate results. Previous works based on machine learning methods mostly used frequency models such as term frequency-inverse document frequency (TF-IDF) [3] as features for sentiment analysis. On the other hand, recent studies use word embeddings as input to the machine learning models. The most commonly used word embeddings in this domain are Word2Vec [4], GloVe [5], and BERT [6].

This paper uses different tokenization methods and embedding types with different machine learning models to extract sentiment from reviews and analyzes which combinations produce better results. We compare the results for the agglutinative language Turkish and the analytical language English. Due to its agglutinative nature, Turkish poses several challenges in NLP studies. The rich morphological structure of the language makes it an open question of how words should be tokenized in different types of tasks. In addition, the free-word order of the sentences brings complications in capturing the semantics of different types of sentences. We include the Turkish language in this work, in addition to the English language, to analyze the effects of these challenging issues in the sentiment analysis domain. Our hypothesis is that the choice of tokenization methods, embedding schemes, and learning models significantly influences the performance of sentiment analysis on various datasets in both agglutinative and analytical languages.

As in recent sentiment analysis studies, we use Word2Vec, GloVe, BERT, and RoBERTa [7] (XLM-RoBERTa [8] for Turkish) embeddings as features. These embeddings can be produced for individual words/tokens or entire documents. We use these embeddings as input to the machine learning models Feed Forward Neural Network (FFNN) and Convolutional Neural Network (CNN) [9]. We use the embedding vectors of BERT and RoBERTa (XLM-RoBERTa) in two ways: for FFNN, we get the embedding vector of the sentence marker ([CLS] token) in the last layer, while for CNN, we get the embedding vectors of all tokens in the last layer.

As a novel approach, we produce embeddings for eleven different tokenization schemes, which are surface form, preprocessed form, surface form with stopwords eliminated, stem, lemma, morphemes, byte pair encoding (BPE) [10], WordPiece [11], unigram language model (ULM) [12], syllables (for Turkish) and partial surface form (for Turkish). Most of the tokenization schemes that we employ in this work are widely used in different types of NLP tasks in the literature [13], [14]. Using surface or root forms of words and some variations of these forms, such as the preprocessed form, is a common approach in the NLP domain [15], [16]. The tokenization methods BPE, WordPiece, and ULM are referred to as subword methods and are used mainly in deep learning-based studies [10], [17]. Besides these widely used schemes, syllables and partial surface forms are not used much as tokenization approaches in the literature. In this work, we make a comprehensive analysis of a large number of tokenization strategies, including those that are commonly used in NLP tasks and those that are not common but may have value in morphologically rich languages.

We use Stanford IMDB Movie Reviews [18], Semeval 2016 Task 5 Restaurant Reviews [19], and Semeval 2017 Task 4 Twitter [20] datasets for English, and Semeval 2016 Task 5 Turkish Restaurant Reviews [19] and Beyazperde Movie Reviews [21] datasets for Turkish. For each dataset, we generate words and subwords based on the tokenization schemes and their embeddings. Then, we train the machine learning models with these embeddings. The analysis of the results shows which models and tokenization types perform better. We also observe that some proposed approaches outperform the current state-of-the-art results. We make the codes publicly available for research purposes.[1]

The main contributions of the paper are as follows:

- We analyze the interactions between different tokenization methods, embedding schemes, and learning models on several datasets in two languages.
- We obtain state-of-the-art results for sentiment analysis on some of the datasets.
- We conduct a stability analysis for the learning models and the tokenization methods.

The rest of the paper is organized as follows: Section II briefly introduces the word embedding approaches, the machine learning models, and the tokenization methods used in this work. Section III presents the previous studies related to sentiment classification. We outline the datasets used in this work in Section IV and describe the proposed models in Section V. The results are given in Section VI. Section VII explains the ensembling of different tokenization methods to increase performance. Finally, Section IX concludes the article.

II. BACKGROUND

A. WORD EMBEDDINGS

Word embeddings are representations of words using vectors that show semantic and syntactic similarities between words. The simplest embedding model is one-hot encoding, where a

[1] https://github.com/alierkan/TokenizedSentimentClassification

vector is created with a size equal to the number of words in the vocabulary.

More efficient word embedding models have been developed and used in recent NLP studies. The most commonly used word embedding models can be cited as Word2Vec [4], GloVe [5], FastText [22], BERT [6], RoBERTa [7], XLM-RoBERTa [8], GPT-2 [23], and GPT-3 [24]. We briefly explain below the models that are used in this work.

1) Word2Vec

Word2Vec is an unsupervised embedding model developed by Mikolov et al. [4]. It has two versions: Skip Gram (SG) and Continuous Bag of Words (CBOW) [25]. The Skip Gram model with negative sampling usually yields more efficient results [26].

2) GloVe

GloVe is another unsupervised model based on word-to-word co-occurrence statistics on which training is performed. The output of training provides linear substructures of the word vector space. It encodes the co-occurrence probability ratio between two words as vector differences. Similarly to Word2Vec, GloVe embeddings give a single embedding for a token independent of its context.

3) BERT

BERT (Bidirectional Encoder Representations from Transformers) is a multilayer bidirectional transformer encoder model that can be used in two ways. Unlike Word2Vec / Love, BERT produces contextual embeddings in the sense that the embedding of a token will be different depending on its context. The first way to use BERT for sentiment classification is, given an input sequence, the embeddings of the words in the sequence can be obtained, and these embedding vectors are used as the feature. As a second way, only the output [CLS] token can be used to classify the sequence. In this work, we make use of both approaches.

BERT has two models: BERT-base and BERT-large. The BERT-base (BERT-large) model consists of an encoder with 12 (16) transformer blocks, 12 (24) self-attention heads, and a hidden size of 768 (1024). The length of the input sequence is limited to 512 tokens.

4) RoBERTa

RoBERTa is a replication of BERT with some modifications. The next sentence prediction objective was removed from the model. The model was trained with larger batch sizes and more data. In addition to the dataset used in BERT for pretraining, Liu et al. [7] used a new large dataset.

5) XLM-RoBERTa

XLM-RoBERTa is a multilingual version of the RoBERTa model. The masked language model was trained in 100 languages using the Common Crawl corpus [27].

B. TOKENIZATIONS

Tokenization is the process of dividing a sentence into tokens. Tokens can be formed of words as written in the sentence, or they can be formed of different forms of words. In this work, we make a comprehensive analysis of the effect of different tokenization approaches explained in the following on the sentiment classification task.

1) WORDS

In this approach, tokens are formed of character sequences separated by a white space. This is the simplest tokenization approach used in NLP tasks. In this model, a token usually corresponds to a word. However, since no preprocessing is applied to the sentence, tokens may correspond to words mixed with punctuation marks or other kinds of characters.

2) WORDS - PREPROCESSED

It is common practice to preprocess sentences or documents before tokenization. In this approach, we use the following preprocessing operations, some of which are tailored to the sentiment analysis domain:

- Lowercase the sentence.
- Change “n’t” to “not.”
- Remove “@name”.
- Isolate and remove punctuations except “?”.
- Remove other non-alphanumeric characters.

3) WORDS - NO STOPWORDS

Stopwords are words that are frequently used in all types of documents in a language and thus do not have discriminative power. In this tokenization approach, we eliminate stopwords in sentences by using the stopword list of the Python NLP library NLTK [28].

4) STEM

Stemming is a rule-based process that strips off suffixes in a word in order to obtain the stem (root form) of the word [29]. In this approach, we use the stems of words as tokens in the sentiment analysis models. For stemming, we use Stanford CoreNLP [30] for English and the morphological parser of Sak et al. [31] for Turkish. Also, we used Sak et al.’s morphological parser in lemmatization and morpheme extraction for Turkish.

5) LEMMA

Lemmatization [32] is similar to stemming in the sense that the root form of the word is extracted. The difference is that it also takes into account morphophonemic rules and irregular cases. Similar to stemming, we use the lemma forms of the words in the models. We use Stanza [33] for the lemmatization in English.

6) MORPHEMES

Similar to the previous two approaches, morphological parsing finds the root form of a word, but also extracts all the suffixes. In this tokenization approach, a word is divided into its morphemes (root and suffixes), and each morpheme is used as a separate token [34], [35]. We use Morfessor [36] for English morpheme extraction.

7) BYTE PAIR ENCODING

Byte pair encoding (BPE) [10] has originated from a data compression algorithm [37]. Its variants are used in Google's SentencePiece tokenization method [38] and OpenAI's GPT-3 model [24]. The intuition behind the BPE algorithm is, given a corpus, to divide the tokens in the corpus into smaller parts (subwords) such that frequently occurring character sequences are represented together. In this approach, we use the subwords as tokens.

8) WordPiece

The WordPiece algorithm [11] is very similar to the BPE tokenization method. The difference is that it combines the sequences that increase the language model probability of the corpus the most.

9) UNIGRAM LANGUAGE MODEL

BPE and WordPiece are bottom-up algorithms that start with individual characters as subwords and form longer subwords by combining consecutive sequences at each iteration. On the other hand, the unigram language model (ULM) [12] is similar to a top-down approach in the sense that it begins with tokens and subwords and eliminates some of these units at each iteration.

10) SYLLABLES

A syllable is a unit of pronunciation that has one vowel sound. Turkish has well-defined rules for syllabification. We use the syllables in the words as tokens.

The Turkish syllabification algorithm is shown in Figure 1 [39]. The algorithm begins by scanning the end of the word and finds the rightmost vowel in the word. If the letter to the left of this vowel is a vowel, then the part of the word from the rightmost vowel up to the end is accepted as a syllable. Otherwise, the part of the word from the consonant on the left up to the end is a syllable. The detected syllable is removed and the algorithm iterates. If no vowel can be found at a step, the remaining word is accepted as a syllable.

11) PARTIAL SURFACE FORM

The partial surface form of a word is an intermediate form between the root form and the surface form [40]. Inspired by this idea, we propose a tokenization method in which the inflectional suffixes are detached from the word, while the derivational suffixes are kept. After the partial surface form and the inflectional suffixes of the word are obtained, each is used as a separate token.

We can explain the intuition behind this method as follows. In morphologically rich languages, using surface forms of words without any segmentation increases the number of words drastically. It causes sparse data problems, while reducing the words to a base form like the stem form causes a loss of information in the suffixes. A remedy to this problem may be to keep the derivational suffixes, which give a semantic shift to the root meaning, and to separate the inflectional suffixes, which do not add to the meaning. In this work, we test this argument in Turkish sentiment analysis models.

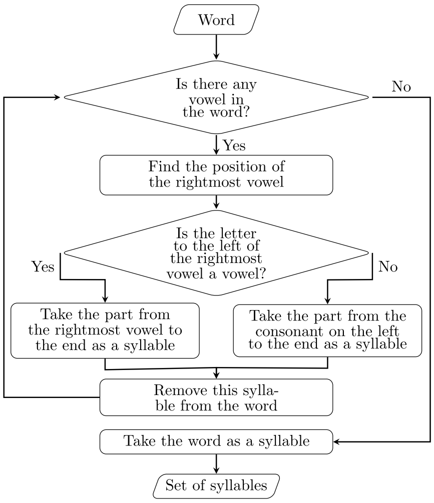
Word
Is there any vowel in the word?
No
Yes
Find the position of the rightmost vowel
Is the letter to the left of the rightmost vowel a vowel?
Yes
No
Take the part from the rightmost vowel to the end as a syllable
Take the part from the consonant on the left to the end as a syllable
Remove this syllable from the word
Take the word as a syllable
Set of syllables

FIGURE 1. Syllable extraction algorithm for Turkish.

Table 1 exemplifies the tokenization methods explained in this section. The first nine methods are shown on a sentence from the IMDB Movie Reviews dataset and the last two are specific to Turkish on a sentence from the BeyazPerde Movie Reviews dataset.

III. LITERATURE REVIEW

Sentiment analysis is a prevalent task in natural language processing and there is plenty of work on this topic. In this section, we only review the recent studies on this subject, especially the studies that use the same datasets as we do. For further reading, Suneetha and Row provide a recent survey of the topic [41].

The approaches used in sentiment analysis can be divided into three: lexicon-based, machine learning-based, and hybrid approaches. Lexicon-based approaches make use of lexicons such as SentiWordNet and WordNet. In machine learning approaches, different methods have been employed, such as support vector machines (SVM) [42], k-nearest neighbors (k-NN) [43], conditional random field (CRF) [44], hidden Markov models (HMM) [45], naïve Bayes (NB) [46], maximum entropy [47], and finally deep learning. Deep learning models based on the LSTM and CNN architectures and their variants currently dominate the field. Hybrid approaches combine lexicon information with machine learning models [48].

There are many surveys related to sentiment analysis. Dang et al. [49] summarized deep learning-based sentiment analysis studies. They analyzed 32 studies that

TABLE 1. Tokenizations of example sentences.

Tokenization Method	Tokens
	Sentence: homelessness (or houselessness as george carlin stated) has been an issue for years
Words	homelessness (or houselessness as george carlin stated) has been an issue for years
Words - Preprocessed	homelessness or houselessness as george carlin stated has been an issue for years
Words - No Stopwords	homelessness houselessness george carlin stated issue years
Stem	homeless or houseless as georg carlin state has been an issu for year
Lemma	homelessness or houselessness as george carlin state have be a issue for year
Morphemes	home less ness (or house less ness as george carlin state d) has been an issue for year s
BPE	home lessness (or house lessness as george carlin stated) has been an issue for years
WordPiece	homeless ness (or house less ness as george carlin stated) has been an issue for years
ULM	home less ness (or house less ness as george car lin stated) has been an issue for years
	Sentence: Kalamardan ve koladan önce balık geldi ve apar topar önümüze atıldı (Fish was served before squid and coke and was thrown in front of us)
Syllables	Ka la mar dan ve ko la dan ön ce ba lık gel di ve a par to par ö nü mü ze a tıl dı
Partial Surface Form	Kalamar dan ve kola dan önce balık gel di ve apar topar ön ümüze atıl dı

use deep neural networks (DNN), CNNs, and recurrent neural networks (RNNs) with Word2Vec embeddings and TF-IDF. Kastrati et al. [50] used CNN and BiLSTM to identify the opinions in Facebook comments related to COVID-19 in the Albanian language. They used FastText and BERT embeddings. Chandra and Krishna [51] employed a BiLSTM model with BERT embeddings in Twitter messages. Hung [52] studied document-level opinion mining using different models such as CNN, LSTM, BiLSTM and CNN-LSTM for the Vietnamese language. He used different tokenization approaches and domain-specific and nondomain-specific embeddings. Shin et al. [53] used lexicon-based embeddings in the CNN model. They proposed three ways of integrating the information from the lexicon into the model. They developed an embedding attention vector to transform the embedding document matrix into a vector. Camacho-Collados and Pilevar [54] analyzed the sentiment classification performance of three different preprocessing techniques, which are lemmatization, lowercasing, and multiword grouping by using the CNN model.

For the IMDB Movie Reviews dataset, Haonan et al. [55] used a pretrained large BERT model for encoding and developed a graph neural network. Each review was represented by a node in the graph. They linked nodes that belong to the same topic, created a graph dataset, and developed the graphstar model architecture, which performs inductive tasks on previously unseen graph data and aggregates local and long-range information. Wang et al. [56] reformulated sentiment classification as a textual entailment task such that an input sentence is combined with user sentiment. They used few-shot learning as a kind of meta-learning by employing a limited number of samples. Meta-learning uses a support set that is a small set of labeled instances instead of a training set, and every class has at most a few samples n (n-shot). The model yielded 96.1 percent accuracy with the complete training set and 87.1 percent accuracy with an 8-shot. Chi et al. [57] used pretrained BERT with fine-tuning and obtained 95.79 percent accuracy. Qizhe et al. [58] proposed the unsupervised data augmentation (UDA) model and used the model with pretrained BERT data. They transformed unlabeled data instances into realistic-looking training data using the UDA model. Table 2 lists the studies related to this dataset.

TABLE 2. Summary of related works on IMDB movie reviews dataset.

Dataset	Feature Set	Model	Accuracy
Haonan et al. (2019) [55]	BERT-Large	Graph Neural Net	96.0
Wang et al. (2021) [56]	RoBERTa-Large	Few-shot learning	96.1
Chi et al. (2019) [57]	BERT-Large	CNN	95.79
Qizhe et al. (2019) [58]	BERT + UDA	BiLSTM	95.8

For the Semeval 2016 Restaurant Reviews dataset, Khalil et al. [59] used Kim's CNN model [9] with Word2Vec embeddings. They used ensemble models with simple voting. Kumar et al. [60] used SVM as the classifier. The features are word polarities retrieved from lexicons, token unigrams and bigrams, and entity-attribute pairs. Brun et al. [61] used a syntactic parser [62] to extract POS tagging, lemma, and surface form of the tokens. The polarities are found by using an ensemble of singular value decomposition (SVD) [63] and elastic net [64]. Wallaart and Frasincar [65] combined domain knowledge in the form of an ontology and a neural rotatory attention model [66] and used GloVe as word embeddings. Trusca et al. [67] extended the model of Wallaart and Frasincaret [65] and used hierarchical attention by adding an extra attention layer to the hybrid approach and replaced GloVe embeddings with ELMo and BERT.

Reddy et al. [68] developed a multi-headed self-attention model with the last five layers of the fine-tuned BERT (BERT-IL). Table 3 summarizes the related works on this dataset. Since it is a small dataset for a specific domain, models with lexicon-based features yielded high accuracies.

TABLE 3. Summary of related works on semeval 2016 restaurant reviews dataset.

Dataset	Feature Set	Model	Accuracy
Khalil et al. (2016) [59]	Word2Vec	CNN	85.49
Kumar et al. (2016) [60]	Lexicon	SVM	86.73
Brun et al. (2016) [61]	Lexicon + CRF	SVD + Elastic Net	88.13
Wallaart et al. (2019) [65]	GloVe	Ontology + LCR-Rot	88.00
Trusca et al. (2020) [67]	ELMo + BERT	Hierarchical Attention	87.00
Natesh et al. (2020) [68]	BERT-IL	Self Attention	88.70

For the Semeval 2017 Twitter dataset, Cliche [69] used CNN and LSTM with GloVe, Word2Vec, and FastText embeddings. He trained CNN and LSTM models separately and then ensembled the output of the two models. Yin et al. [70] employed Recurrent-CNN (RCNN) with GloVe and Word2Vec embeddings. RCNN integrates recurrent networks with cohesive convolutional models [71]. Hamdan [72] used CNN with structured skip-gram embeddings. Baziotis et al. [73] used bidirectional LSTM (BiLSTM) with GloVe and Word2Vec embeddings. Table 4 summarizes the related studies and shows their accuracies.

TABLE 4. Summary of related works on semeval 2017 twitter dataset.

Dataset	Feature Set	Model	Accuracy
Cliche (2017) [69]	Word2Vec GloVe FastText	CNN + LSTM	65.8
Yin et al. (2017) [70]	Word2Vec GloVe	RCNN	66.4
Hussam (2017) [72]	Word2Vec	CNN	65.2
Baziotis et al. (2017) [73]	Word2Vec GloVe	BiLSTM	65.1

For the Semeval 2016 Turkish Restaurant Reviews dataset, similar to their experiments on the English Restaurant Reviews dataset, Kumar et al. [60] used word polarities, token n-grams, and entity-attribute pairs as features in an SVM classifier. Ruder et al. [74] used CNN [9] for aspect-based sentiment analysis. They first extracted the aspect tokens and then fed the embeddings of the aspects and words together to the CNN model. Table 5 summarizes the related works on this dataset. Similar to the English counterpart of this dataset, the lexicon-based model yields high performance.

TABLE 5. Summary of related works on semeval 2016 turkish restaurant reviews dataset.

Dataset	Feature Set	Model	Accuracy
Kumar et al. (2016) [60]	Lexicon	SVM	84.23
Ruder et al. (2016) [74]	GloVe	CNN	74.21

Finally, for the Beyazperde Movie Reviews dataset, Uçan et al. [21] used a lexicon and SVM to predict the sentiments of the reviews. They obtained 84.6 percent accuracy. Aydın et al. [75] combined Word2Vec with lexicon-based polarity scores to obtain features and used SVM as the classifier. In another work, Aydın and Güngör [40] used SVM with the so-called partial surface forms of the words. They combined an unsupervised model with the supervised model using majority voting. For the unsupervised model, they used the sentiment score of a word based on pointwise mutual information (PMI) [76] to decide the sentiment. For the supervised model, they used TF-IDF and a neural network model. The related works on this dataset are summarized in Table 6.

TABLE 6. Summary of related works on beyazperde movie reviews dataset.

Dataset	Feature Set	Model	Accuracy
Ucan et al. (2016) [21]	Lexicon-based	SVM	84.6
Aydın et al. (2020) [75]	Lexicon-based + Word2Vec	SVM	90.38
Aydın et al. (2021) [40]	Lexicon + TFIDF	PMI + Neural Network	91.17

There are also several other sentiment analysis studies for the Turkish language. Kaya et al. [77] used maximum entropy, n-gram language model, SVM, and naïve Bayes to classify sentiments of Turkish political columns. Çetin and Amasyali [78] employed active learning for sentiment analysis on Turkish tweets. Türkmenoğlu and Tantuğ [79] translated an English sentiment lexicon to Turkish and employed both lexicon-based and machine learning-based methods. Vural et al. [80] customized the SentiStrength sentiment analysis library by translating its lexicon to Turkish and using an unsupervised learning model. Dehkharghani et al. [81] created SentiTurkNet, a Turkish polarity lexicon for sentiment analysis.

To provide a summary of the previous works covered in this section, we show in Table 7 the embedding methods/features and the learning models used in these works.

TABLE 7. List of embedding methods/features and the learning models used in related studies.

Embedding methods or features	Learning models
Word2Vec	SVM
Glove	LSTM, BiLSTM
FastText	CNN, RCNN
BERT	Graph Neural Network
RoBERTa	Self and Hierarchical attention
ELMo	
Lexicon	
TF-IDF	

IV. DATASETS

In this work, we use three datasets in English and two datasets in Turkish to analyze the effects of different model architectures and tokenization methods on both languages.

TABLE 8. Number of reviews in the datasets.

	Training			Test		
Dataset	Positive	Negative	Neutral	Positive	Negative	Neutral
IMDB Movie Reviews [18]	12500	12500	–	12500	12500	–
Semeval 2016 Restaurant Reviews [19]	1478	513	57	510	139	27
Semeval 2017 Twitter [20]	19902	7840	22591	2375	3972	5937
Semeval 2016 Turkish Restaurant Reviews [19]	742	469	91	105	39	4
Beyazperde Movie Reviews [21]	13350	13350	–	13350	13350	–

The English datasets are the Stanford IMDB Movie Reviews dataset [18], Semeval 2016 Task 5 Restaurant Reviews dataset [19], and the Semeval 2017 Task 4 Twitter dataset [20]. The Turkish data sets are the Semeval 2016 Task 5 Restaurant Reviews dataset [19] and the Beyazperde Movie Reviews dataset [21]. Table 8 shows the number of reviews in the sentiment classes in the datasets. The movie review datasets (IMDB and Beyazperde) are two-class datasets and are balanced. The Semeval Restaurant Reviews and Twitter datasets, on the other hand, are skewed datasets formed of three sentiment classes. We used the train and test splits of the datasets shown in Table 8 in order to be able to compare our results with the results in the literature. Additionally, to avoid fitting the models to the specifics of the test sets and to generalize the results, we employed k-fold cross-validation (k = 5) for all the datasets, learning models, and tokenization methods.

We evaluate the results in terms of accuracy. For unbalanced datasets, we also give weighted averaged F1 value scores. The weighted-averaged F1-value is computed by averaging the F1-values of the classes while considering the support of each class. The support of a class is the number of instances in the dataset that belong to that class. F1-value of a class is calculated using Equation 1, where TP, FP, and FN correspond to true positives, false positives, and false negatives, respectively.

$$\mathrm{F1} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2} * (\mathrm{FP} + \mathrm{FN})} \tag{1}$$

V. MODELS

In this work, we use two machine learning models: feed-forward neural network (FFNN) and convolutional neural network (CNN). We also use two different input representations. The first one is the combination of Word2Vec and GloVe embedding vectors, and the second one is pretrained BERT and RoBERTa (XLM-RoBERTa) embeddings. Thus, we build different models, which are explained in the subsections below.

As stated in Section II-B, we employ eleven different tokenization methods (Words, Words-Preprocessed, Words-No Stopwords, Stem, Lemma, Morphemes, Byte pair encoding, WordPiece, Unigram language model, Syllables, Partial surface form) and compare their effects on sentiment analysis. Each model is experimented with these tokenization schemes and the results are given in Section VI. Although we use pre-trained BERT and RoBERTa (XLM-RoBERTa) embeddings with these tokenizations in our models, we train Word2Vec and GloVe embeddings from scratch with these tokenizations and use them.

A. CNN WITH Word2Vec AND GloVe EMBEDDINGS

We generate Word2Vec and GloVe embeddings from our datasets. We also use the YELP dataset [82] to generate embeddings for the Semeval 2016 restaurant dataset. Figure 2 shows the CNN model used in this work. The model consists of one convolution layer, one max-pooling layer, and one feed-forward layer. The rows in the two-dimensional input tensor correspond to the words in the review. We use padding and set the number of rows N to the number of words in the longest review in the dataset. Words have dimension $d_1 + d_2$, where d_1 and d_2 are, respectively, Word2Vec and GloVe embedding dimensions of the word. We concatenate the Word2Vec embedding and the GloVe embedding of the word.

M is the number of filters applied at the convolution layer. In the max-pooling layer, the maximum value of each filter output is taken to form the first hidden layer. There are two hidden layers. We use the rectified linear unit (ReLU) function for the max-pooling layer and the first hidden

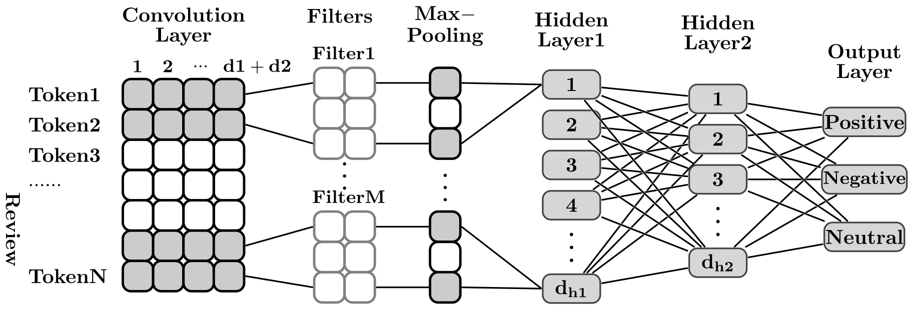

FIGURE 2. CNN model with Word2Vec and GloVe embeddings.

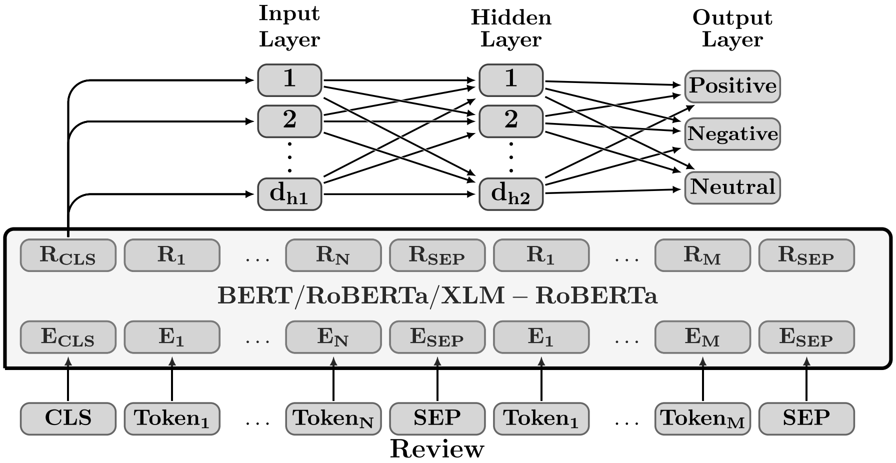

FIGURE 3. FFNN model with Bert/RoBERTa/XLM-RoBERTa embeddings.

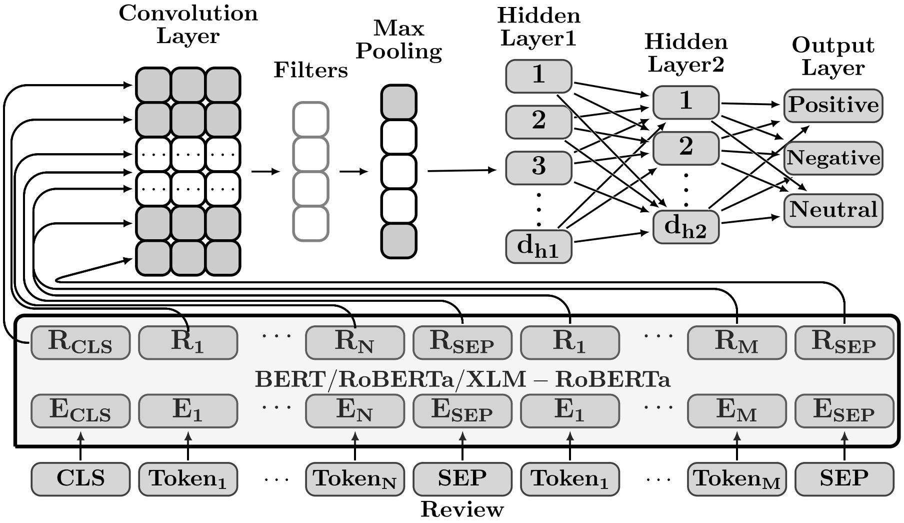

FIGURE 4. CNN model with Bert/RoBERTa/XLM-RoBERTa embeddings.

layer. We use sigmoid for two-class datasets and softmax for three-class datasets for the output layer. The dropout after the max-pooling layer is set to 0.20. As the optimizer, we use adaptive learning rate (Adadelta) [83]. We use CNN with static and dynamic embeddings. In the static CNN model (CNN Static), initial embedding vectors are not updated during training, whereas in the dynamic CNN model (CNN Dyn), initial embedding vectors are updated according to the learned weights during backpropagation.

B. BERT/RoBERTa/XLM-RoBERTa AND FFNN

In this model, we use BERT and RoBERTa (XLM-RoBERTa) as a classifier and combine its output with FFNN. The review is given as input to the BERT/RoBERTa/

XLM-RoBERTa-base model, which comprises 12 layers with an embedding size of 756. The [CLS] token of the last layer that is used for classification is taken and given to an FFNN model with one hidden layer. In this way, we can represent every review by one embedding vector. In the output layer, sigmoid/softmax is applied as the activation function to get the classification decision. The model is shown in Figure 3.

C. CNN WITH BERT/RoBERTa/XLM-RoBERTa EMBEDDINGS

This model is the same as the model in Section V-A, with the difference that BERT and RoBERTa (XLM-RoBERTa) embeddings are used instead of Word2Vec and GloVe embeddings. Figure 4 shows the combined model. The review is given as input to the BERT/RoBERTa/XLM-RoBERTa-base model and output vectors of all tokens at the last layer are used to construct a two-dimensional convolution tensor instead of using only the [CLS] token of the last layer. Following the convolution layer, similar to the previous CNN model, we apply filters and continue with the max-pooling layer, two hidden layers, and the sigmoid/softmax layer. We use the dynamic CNN model.

VI. EXPERIMENTS

In this section, we show the performance of the models explained in Section V on five datasets. For ease of visualization, in the tables given in this section, we group the results concerning the embedding type or model (Word2Vec, GloVe, BERT, RoBERTa, XLM-RoBERTa), underlying the classification model and then the primary classification model (FFNN, CNN). For each model, the results of all the tokenization methods are depicted. We give both the results on the standard train/test splits of the datasets and the results obtained with k-fold cross-validation (k = 5).

TABLE 9. Accuracy results for stanford IMDB movie reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Words	89.54	90.41	**93.26**	93.12	**93.19**	92.88
Words-Preprocessed	**90.66**	**91.01**	90.49	92.93	93.10	**93.31**
Words-No Stopwords	90.21	90.67	89.43	90.57	91.78	91.86
Stem	90.50	90.94	72.51	88.55	91.50	90.98
Lemma	89.96	90.96	92.32	91.75	93.14	92.49
Morphemes	89.83	90.30	91.91	91.45	92.20	92.08
BPE	90.39	90.39	92.68	92.60	93.00	92.20
WordPiece	90.46	90.30	92.54	**93.16**	92.60	90.83
ULM	90.35	90.87	90.56	82.97	91.85	85.39

Tables 9 and 10 show the results of Stanford IMDB movie reviews. The results on the left under the title "Word2Vec+GloVe" correspond to the CNN model where Word2Vec and GloVe embeddings are used (Section V-A). The results on the right of the table correspond to the FFNN (Section V-B) and CNN (Section V-C) models where BERT-base and RoBERTa-base (XLM-RoBERTa-base in Turkish dataset experiments) embeddings are used. The highest accuracy result for each model is shown in bold. CNN models with Word2Vec and GloVe embeddings and BERT-/RoBERTa-based models show mostly the best performance with the surface forms or the preprocessed forms of the words. The use of RoBERTa embeddings in the dynamic CNN model for the standard train/test split and in the FFNN model for k-fold cross-validation gives the overall best result in this dataset. The results in Table 9 are also visualized in Figure 5 to facilitate comparisons with respect to tokenization methods and models.

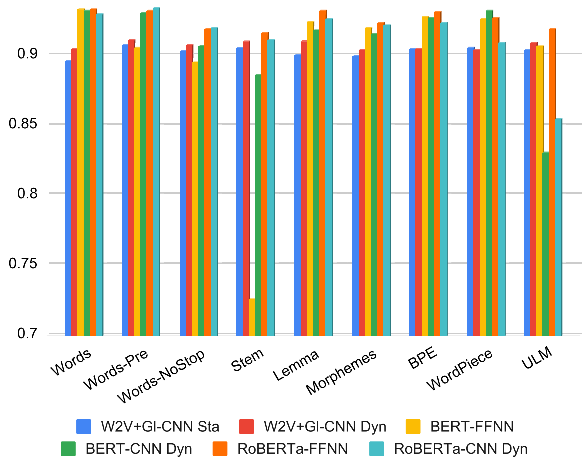

FIGURE 5. Accuracy results for stanford IMDB movie reviews dataset.

TABLE 10. K-fold accuracy results for stanford IMDB movie reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Words	**91.32**	**91.09**	**93.99**	**93.36**	95.06	94.26
Words-Preprocessed	90.04	89.72	93.44	92.99	95.00	94.46
Words-No Stopwords	89.19	89.60	90.63	90.66	93.26	92.52
Stem	87.40	89.22	90.16	90.26	94.35	93.69
Lemma	89.35	90.54	91.31	92.74	94.78	94.22
Morphemes	87.05	88.90	91.31	91.07	94.53	93.63
BPE	89.05	89.68	93.75	93.24	94.97	**94.80**
WordPiece	87.29	90.46	93.87	93.08	**95.17**	94.47
ULM	89.36	89.98	91.10	91.60	94.40	94.06

In Word2Vec and GloVe stem and lemma tokenizations and BERT/RoBERTa, BPE and WordPiece tokenizations also achieve comparable results. Compared to the cases where all words are used (Words and Words-Preprocessed), stopword elimination seems to degrade the performance. This indicates that some stopwords carry sentiment meaning and eliminating them causes loss of sentiment. ULM tokenizations with BERT/RoBERTa CNN models and stemming with BERT models show low performance in the dataset. Finally, the comparison of Word2Vec/GloVe-based and BERT/RoBERTa-based models shows that the BERT/RoBERTa-based models outperform by 2.30 points/3.85 points (standard split/k-fold cross-validation) and RoBERTa-based models are more stable than the BERT-based models.

TABLE 11. Accuracy results for semeval 2016 restaurant reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	**CNN Static**	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Word	83.45	84.20	**86.07**	**85.87**	**86.03**	**85.25**
Word-Preprocessed	**84.35**	**84.67**	84.56	85.70	**86.44**	85.06
Word-No Stopwords	84.11	83.71	83.45	83.61	71.89	81.95
Stem	83.96	84.63	85.27	85.21	82.28	83.43
Lemma	83.30	83.89	85.17	84.57	85.41	83.21
Morphemes	79.51	79.74	79.63	80.08	80.06	52.96
BPE	80.40	80.52	72.86	71.21	75.44	75.44
WordPiece	**84.53**	**84.76**	**86.30**	**85.65**	85.21	**85.53**
ULM	81.67	83.20	73.59	64.70	75.44	76.78

Table 11 shows the accuracy of the models for Semeval 2016 Restaurant Reviews. The results are similar to those in the IMDB dataset in the sense that preprocessed words in Word2Vec/GloVe models, raw words in BERT models, and these two tokenizations in RoBERTa models yield successful results. However, the WordPiece tokenization method also gives success rates similar to those of the word-based methods in most cases. To see this effect clearly, we marked more than one result as bold in the table columns. ULM and BPE tokenizations with BERT/RoBERTa models get the worst accuracies.

Tables 12 and 13 and Figure 6 (corresponding to Table 12) show the macro F1-value scores of the models. Since Semeval 2016 Restaurant Reviews is a skewed dataset, the F1-values are lower than the accuracy values. The F1-values exhibit similar behavior as the accuracy results. However, the best F1-values on the standard train/test split are obtained with RoBERTa embedding and stem tokenization.

The results for the Semeval 2017 Twitter dataset are shown in Tables 14, 15, and 16 and in Figure 7 (corresponding to Table 15). This dataset differs from the other datasets in the sense that it is more equally distributed over three classes. Hence, the F1-values are close to accuracy values. In Word2Vec and GloVe models, stem forms and ULM subwords yield the top results, and the lemma forms obtain comparable results.

TABLE 12. Macro F1-values for semeval 2016 restaurant reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	**CNN Static**	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Word	59.19	65.80	**68.61**	66.56	70.72	53.81
Word-Preprocessed	**62.23**	66.93	66.65	**68.26**	55.89	55.23
Words-No Stopwords	59.25	63.28	63.35	60.85	66.31	66.61
Stem	57.06	66.54	66.35	65.07	**71.43**	**67.24**
Lemma	59.40	64.26	66.80	64.60	54.60	54.78
Morpheme	48.86	50.30	50.92	51.51	54.40	44.90
BPE	52.83	58.75	44.76	43.53	55.74	56.50
Wordpiece	60.92	**67.71**	66.98	66.79	55.72	53.29
ULM	52.84	62.08	41.35	24.52	53.61	60.88

TABLE 13. K-fold macro F1-values for semeval 2016 restaurant reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	**CNN Static**	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Words	51.26	60.17	55.51	55.23	56.54	56.95
Words-Preprocessed	51.26	62.53	**56.00**	**56.06**	57.02	56.96
Words-No Stopwords	53.19	57.93	52.77	51.94	51.04	52.11
Stem	53.45	61.98	52.84	51.87	49.65	52.58
Lemma	52.54	61.21	55.46	55.50	54.26	56.21
Morphemes	44.42	50.19	45.52	45.43	44.74	38.40
BPE	48.97	58.55	28.15	28.56	28.15	56.62
WordPiece	**53.97**	**64.39**	55.46	55.36	56.54	56.04
ULM	52.49	57.37	54.13	54.01	**63.35**	**64.26**

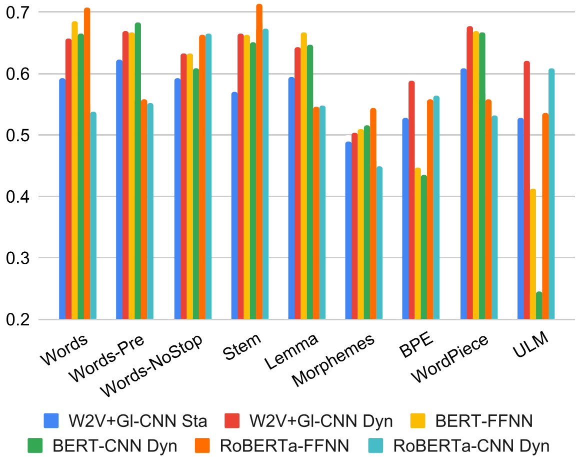

FIGURE 6. Macro F1-values for semeval 2016 restaurant reviews dataset.

In the case of BERT/RoBERTa models, the raw forms of the words again show the best performance. The

TABLE 14. Accuracy results for semeval 2017 twitter dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Word	55.76	56.08	**67.14**	**67.19**	**70.66**	**69.40**
Word-Preprocessed	56.26	56.08	66.44	66.38	67.34	68.52
Word-No Stopwords	55.55	55.16	63.67	64.30	67.71	66.48
Stem	**57.31**	57.16	64.56	64.11	68.44	65.61
Lemma	57.03	56.87	63.09	62.68	67.57	65.79
Morphemes	52.21	52.88	55.16	41.09	58.45	49.96
BPE	56.46	56.95	48.33	48.33	64.35	61.27
WordPiece	56.70	56.54	65.74	65.86	69.57	67.07
ULM	56.92	**57.25**	48.33	48.33	66.31	66.32

TABLE 15. Macro F1-values for semeval 2017 twitter dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Word	55.35	55.91	**66.65**	**66.12**	**70.80**	**69.82**
Word-Preprocessed	56.12	55.56	66.13	66.00	69.28	67.18
Word-No Stopwords	55.23	54.82	63.08	63.60	65.79	65.38
Stem	**57.43**	**57.37**	63.80	63.60	65.79	65.38
Lemma	55.53	56.02	62.30	61.86	63.94	63.01
Morphemes	45.30	46.41	48.19	41.32	52.29	43.72
BPE	55.76	56.47	21.72	16.29	63.71	63.03
WordPiece	56.67	56.53	65.44	65.47	69.29	65.65
ULM	56.36	57.07	21.72	19.01	66.33	63.27

TABLE 16. K-fold macro F-1 values for semeval 2017 twitter dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Words	57.05	58.79	**69.87**	**70.39**	**70.14**	**70.39**
Words-Preprocessed	57.05	58.59	68.98	69.11	68.41	69.11
Words-No Stopwords	54.79	57.36	66.59	66.87	66.67	66.87
Stem	**58.64**	59.14	67.55	67.64	68.60	67.64
Lemma	56.38	58.95	68.91	69.09	69.15	67.83
Morphemes	45.10	45.81	51.06	50.43	51.64	49.53
BPE	57.40	59.03	30.33	47.62	64.37	47.62
WordPiece	56.71	59.24	68.51	68.96	67.53	68.96
ULM	56.99	**59.59**	68.93	67.83	64.99	65.99

RoBERTa-based models achieve state-of-the-art results using the standard train/test split of the dataset. This indicates that pre-trained RoBERTa embeddings may be more suitable than BERT embeddings for non-domain-specific datasets, an argument that should be tested with other general domain datasets.

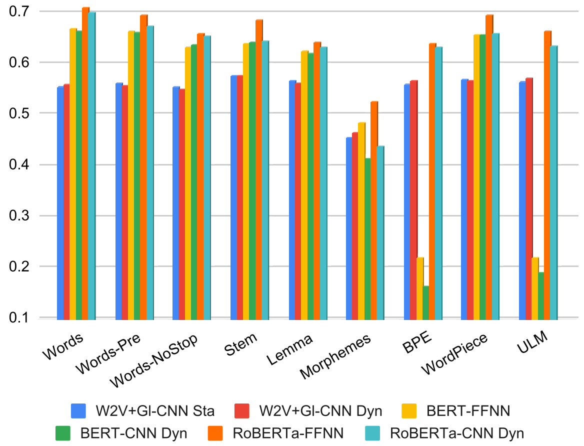

FIGURE 7. Macro F1-values for semeval 2017 twitter dataset.

TABLE 17. Accuracy results for semeval 2016 turkish restaurant reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		XLM-RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Words	71.49	**77.84**	82.77	83.92	82.57	82.94
Words-Preprocessed	72.16	**77.84**	83.11	84.19	**84.32**	**83.51**
Words-No Stopwords	71.89	76.35	**84.12**	83.11	81.62	80.54
Stem	70.54	71.49	78.04	81.08	77.70	80.74
Lemma	71.28	75.54	81.08	82.70	80.54	80.14
Morphemes	69.05	72.57	79.73	81.22	78.65	81.22
BPE	70.95	72.13	73.65	75.23	71.22	71.96
WordPiece	70.54	72.30	82.09	**84.73**	80.41	78.89
ULM	70.95	72.07	65.54	64.86	79.59	81.42
Syllables	**73.11**	76.52	75.00	77.03	71.22	67.57
Partial S.F.	67.84	74.19	79.32	80.27	66.70	67.34

In addition to the three English datasets, we give the results for the two Turkish datasets. Tables 17, 18, and 19 and Figure 8 (corresponding to Table 18) show the results for the Semeval 2016 Turkish Restaurant Reviews dataset, and Tables 20 and 21 and Figure 9 (corresponding to Table 20) show the results for the BeyazPerde Movie Reviews dataset. Since no pre-trained RoBERTa model exists for Turkish, we used the pre-trained XLM-RoBERTa embeddings. Different from the English datasets, the tables and the figures also include the results for the two Turkish-specific tokenization methods, which are syllables and partial surface forms. In the Semeval 2016 Turkish Restaurant Reviews

TABLE 18. Macro F1-values for semeval 2016 turkish restaurant reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		XLM-RoBERTa-base	
Tokenizations	**CNN Static**	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Words	29.74	48.73	64.00	63.80	**61.10**	59.60
Words-Preprocessed	34.49	49.72	63.69	**64.72**	54.82	**61.48**
Words-No Stopwords	31.79	**54.26**	**65.31**	62.29	53.24	63.86
Stem	29.87	45.04	57.44	58.87	56.92	55.96
Lemma	28.02	44.08	59.21	63.17	55.41	55.41
Morphemes	**41.31**	43.98	57.56	61.11	49.59	57.68
BPE	27.67	33.68	51.84	33.42	42.16	49.26
WordPiece	28.15	45.35	62.20	63.85	51.80	53.01
ULM	27.67	33.98	36.72	31.29	51.66	51.66
Syllables	39.06	44.17	54.18	46.60	27.67	44.42
Partial S.F.	34.15	47.78	54.62	60.69	39.44	56.92

TABLE 19. K-fold macro F-1 values for semeval 2016 turkish restaurant reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		XLM-RoBERTa-base	
Tokenizations	**CNN Static**	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Words	25.41	46.61	55.98	56.23	56.98	60.57
Words-Preprocessed	25.41	**47.61**	**57.10**	55.69	55.07	**63.19**
Words-No Stopwords	31.01	47.45	55.33	**56.97**	56.63	60.80
Stem	25.43	43.86	53.21	51.87	53.82	59.66
Lemma	24.72	43.82	54.73	53.76	53.57	61.30
Morphemes	**34.81**	43.68	52.76	52.95	53.86	61.41
BPE	24.58	30.95	24.58	29.80	43.35	38.93
WordPiece	25.13	40.86	53.62	52.20	**57.15**	62.46
ULM	24.58	34.61	53.63	53.44	52.91	61.92
Syllables	29.91	35.16	35.52	35.16	43.55	49.03
Partial S.F.	27.48	42.53	52.73	50.31	51.47	56.93

dataset, the surface forms and the preprocessed forms of the words show high performance as in the English datasets and yield the highest performance in some models. The results obtained by morpheme and syllable tokenizations in Word2Vec and GloVe models and the results obtained by stopword elimination and WordPiece tokenization in BERT models are among the best results.

In Beyazperde Movie Reviews, preprocessed words in Word2Vec and GloVe models and raw words in BERT and XLM-RoBERTa models achieve the highest accuracy values. WordPiece tokenization obtains comparable results in BERT and XLM-RoBERTa models. In addition to surface form and WordPiece tokenizations, the XLM-RoBERTa-based model achieves high accuracies with ULM tokenization.

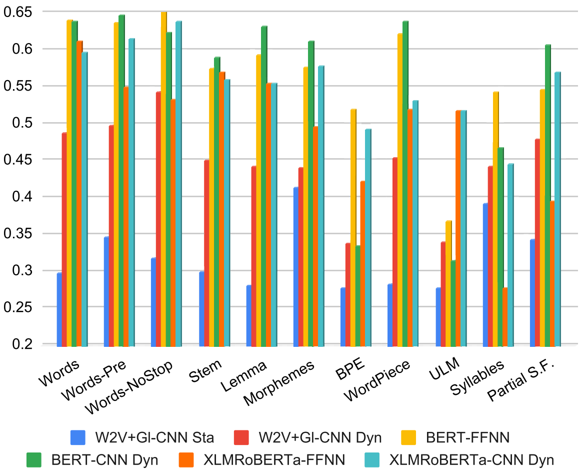

FIGURE 8. Macro F1-values for semeval 2016 turkish restaurant reviews dataset.

In the two Turkish datasets, we also tested the effects of two novel tokenization approaches, syllables and partial surface forms, that might contribute to tokenization in morphologically rich languages. However, except in a few cases, the results indicate that they do not add to the performance of the models, at least in the sentiment analysis domain.

TABLE 20. Accuracy results for BeyazPerde movie reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		XLM-RoBERTa-base	
Tokenizations	**CNN Static**	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Words	91.48	91.42	94.37	**94.20**	**93.90**	**93.81**
Words-Preprocessed	**92.39**	**92.64**	94.21	**94.20**	93.52	93.27
Words-No Stopwords	92.34	91.60	93.61	92.73	92.64	93.31
Stem	88.79	89.86	90.48	80.60	91.10	91.24
Lemma	90.68	90.81	93.21	92.82	92.29	93.18
Morphemes	90.58	90.81	75.52	92.00	91.67	92.17
BPE	90.80	91.03	66.75	63.22	90.63	86.99
WordPiece	90.83	90.89	**94.39**	93.49	93.57	93.53
ULM	91.27	91.42	93.83	93.46	93.43	93.58
Syllables	89.96	90.09	83.71	86.59	87.89	90.91
Partial S.F.	90.64	90.54	92.44	88.77	91.57	91.92

As the results of the experiments on datasets in different domains and two different languages, we observe a general pattern. When non-contextualized embeddings like Word2Vec and GloVe are used, preprocessing the words with a few preprocessing operations or taking the stem of the words increases the success rates. These tokenization approaches outperform the popular subword methods BPE,

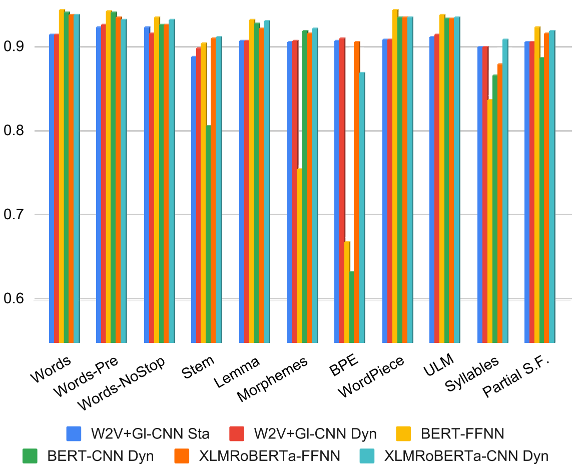

FIGURE 9. Accuracy results for BeyazPerde movie reviews dataset.

TABLE 21. K-fold Accuracy results for BeyazPerde movie reviews dataset.

Embeddings:	Word2Vec + GloVe		BERT-base		XLM-RoBERTa-base	
Tokenizations	CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
Words	87.44	90.04	**94.37**	91.11	93.60	93.29
Words-Preprocessed	88.20	**90.22**	94.24	91.11	**93.61**	92.02
Words-No Stopwords	88.61	89.70	93.67	81.22	92.90	90.28
Stem	87.70	88.79	91.75	88.29	91.34	91.36
Lemma	88.43	88.59	93.37	90.90	92.29	92.75
Morphemes	89.08	89.21	93.79	71.44	93.12	93.07
BPE	88.44	64.29	90.45	84.48	91.28	90.87
WordPiece	88.98	89.11	94.27	**92.05**	93.46	**93.49**
ULM	87.11	90.12	93.57	90.35	92.28	93.24
Syllables	**89.37**	88.94	86.64	90.37	90.03	89.34
Partial S.F.	88.70	87.54	92.46	91.52	91.84	90.59

WordPiece, and ULM. When the contextualized embedding methods BERT and RoBERTa (XLM-RoBERTa) are used, feeding raw words as input or applying WordPiece tokenization works well. Preprocessing the words yields comparable results. The high success of raw words compared to more complex tokenizations can be attributed to BERT/RoBERTa (XLM-RoBERTa) being more powerful models that are built upon the transformer architecture and to the contextualized nature of the embeddings. Furthermore, the success of WordPiece tokenization is probably due to the use of the same tokenization method in the BERT model. Comparing the two embedding approaches, non-contextualized and contextualized embeddings, the latter outperforms the former by a large margin. Finally, we observe that the accuracy results obtained in the Semeval 2017 Twitter dataset (70.66), the Semeval 2016 Turkish Restaurant Reviews dataset (84.73) and the BeyazPerde Movie Reviews dataset (94.39) on the standard train/test splits used in the literature are state-of-the-art results.

TABLE 22. Stability analysis of models over tokenization methods. The maximum and average accuracies and the standard deviation are shown for each model and dataset.

Datasets	Statistics	Word2Vec + GloVe		BERT-base		RoBERTa-base	
		CNN Static	CNN Dyn	FFNN	CNN Dyn	FFNN	CNN Dyn
IMDB	Max	90.66	91.01	93.26	93.16	93.19	**93.31**
	Avg	90.21	90.65	89.52	90.79	92.48	91.34
	Std.Dev.	0.36	0.30	6.50	3.29	0.67	2.37
Semeval Restaurant	Max	84.53	84.76	86.30	85.87	**86.44**	85.53
	Avg	82.81	83.26	81.88	80.73	80.91	78.11
	Std.Dev.	1.84	1.86	5.29	7.63	5.47	10.24
Semeval Twitter	Max	57.31	57.25	67.14	67.19	**70.66**	69.40
	Avg	56.02	56.11	60.27	58.70	66.71	64.49
	Std.Dev.	1.54	1.38	7.62	9.90	3.59	5.91
English	Avg. of Std.Dev.	1.25	1.18	6.47	6.94	3.24	6.17
Datasets	**Statistics**	**Word2Vec + GloVe**		**BERT-base**		**XLM-RoBERTa-base**	
		CNN Static	**CNN Dyn**	**FFNN**	**CNN Dyn**	**FFNN**	**CNN Dyn**
Semeval Restaurant Turkish	Max	73.11	77.84	84.12	**84.73**	84.32	83.51
	Avg	70.89	74.44	78.59	79.85	77.69	77.84
	Std.Dev.	1.45	2.45	5.45	5.79	5.55	5.96
Beyaz perde	Max	92.39	92.64	**94.39**	94.20	93.90	93.81
	Avg	90.89	91.01	88.41	88.37	92.02	92.17
	Std.Dev.	1.02	0.76	9.29	9.32	1.76	1.98
Turkish	Avg. of Std.Dev.	1.24	1.61	7.37	7.56	3.66	3.97

In addition to the performance analysis of the tokenization methods and neural models, we make an analysis of the stability of the learning models over the tokenization methods.[2] The analysis shows how much each model is affected by a change in the tokenization method. Table 22 shows the maximum and average accuracies and the standard deviation for each model per dataset. For each dataset, the highest of the maximum accuracy values is shown in bold. When we compare the standard deviations of the models, we see that the CNN models based on Word2Vec and GloVe embeddings are much more stable than the BERT and RoBERTa (XLM-RoBERTa) models in all the datasets for both languages. This indicates that a change in the tokenization method when we employ non-contextualized embeddings does not affect the performance much. Among the BERT-based embeddings, RoBERTa and XLM-RoBERTa mostly yield more stable results independent of the primary model being a

[2] In the rest of the paper, we will use the accuracy results obtained on the standard train and test splits of the datasets only for ease of the discussion.

TABLE 23. Stability analysis of tokenization methods over models. The standard deviation averaged over the models is shown for each model and English/Turkish datasets.

Tokenization	English Std. Dev.	Turkish Std. Dev.
Words	4.48	3.72
Words-Preprocessed	3.80	3.43
Words-No Stopwords	5.21	3.30
Stem	6.58	5.26
Lemma	3.16	3.35
Morphemes	9.03	6.97
BPE	6.02	9.08
WordPiece	3.63	4.40
ULM	8.94	5.40
Syllables	—	3.98
Partial Surface Form	—	4.98

feed-forward or convolutional network. In other words, the performance of BERT models is affected more than those of the RoBERTa and XLM-RoBERTa models by a change in the tokenization method. Finally, a comparison of each model with respect to different datasets shows that nearly all models are more stable in two-class review datasets than the three-class Semeval datasets.

It may also be interesting to observe how much tokenization methods are affected by a change in the learning model. In a similar manner, we make an analysis of the stability of the tokenization methods over the learning models. Table 23 shows the standard deviation averaged over the models and the English/Turkish datasets for each method. We see that using words without any tokenization (surface forms, preprocessed forms, surface forms with stopwords eliminated), lemma tokenization, WordPiece tokenization, and (for Turkish) syllable tokenization are more robust in the case of a model change. Lemmatization is the most stable tokenization method, indicating that a change in the learning model does not affect much the classification performance. On the other hand, the performance is highly dependent on the learning model when stems, morphemes, BPE subwords, or ULM subwords are used as tokens.

Table 24 shows some example reviews that are predicted incorrectly. Especially in cases when the review includes words with opposing sentiments or negation words or affixes, the models make wrong predictions. For instance, in the sentence “Nice ambience, but highly overrated place,” the word “nice” has a positive sentiment, while the word “overrated” has a negative sentiment. Similarly, the Turkish sentence “Bu fiyata böyle bir memnuniyet çok az rastlanacak şey” includes a positive word “memnuniyet” (satisfaction) and a negative phrase “çok az” (very little). Therefore, when opposing emotions are involved in a review, the models are challenged and may make incorrect predictions. The other examples in the table also reflect this situation. Also, since neutral samples are very few in both training and test splits, the models had trouble making accurate predictions for neutral samples.

VII. ENSEMBLE

Ensembling the results of different machine learning approaches is a commonly used technique to increase performance, which enables the models to compensate for the weaknesses of each other. In this respect, we ensemble the results of the models with different tokenization methods. We take the most successful combinations of the tokenization methods and the models in each dataset given in Tables 9-21. We first used max voting as the ensemble approach, where simply the class with the highest number of votes of the classifiers is chosen. However, we observed that the results were not as high as expected due to the limitations of the max voting strategy [84].

The max voting strategy assigns equal weights to all the classifiers, which can dilute the contribution of stronger models and hinder the ensemble’s performance. In addition, when the individual models are too similar or highly correlated, the ensemble cannot achieve a significant performance improvement. Another limitation is that it is sensitive to noisy predictions produced by the individual models.

To solve these problems, rather than just picking the class with the highest vote, we follow a different strategy by feeding the class decisions of the classifiers to a feed-forward neural network. We use a network with two hidden layers. The dimension of the first layer is equal to the number of classifiers ensembled, and the dimension of the second layer is half of that of the first layer. We use the ReLU activation function for the hidden layers and softmax for the output layer. We use Adadelta as an optimizer. This strategy provides the ensembler to adapt itself to the contribution of each classifier.

Table 25 shows the tokenization methods used for each model and the accuracy results. For instance, in the IMDB Movie Reviews dataset, a total of 11 model and tokenization method combinations (three tokenizations with Word2Vec+GloVe embeddings and dynamic CNN, two tokenizations with BERT embeddings and FFNN, two tokenizations with BERT embeddings and dynamic CNN, two tokenizations with RoBERTa embeddings and FFNN, and two tokenizations with RoBERTa embeddings and dynamic CNN) are ensembled. The value in parenthesis in the last column denotes the increase over the best result for those datasets given in Tables 9-21. We see an increase for all the datasets. In addition to the state-of-the-art results for the Semeval 2017 Twitter, Semeval 2016 Turkish Restaurant Reviews, and BeyazPerde Movie Reviews datasets, the ensemble results for the IMDB Movie Reviews and Semeval 2016 Restaurant Reviews datasets get close to the state-of-the-art. We see that, in addition to the surface forms of the words, different tokenization methods contribute to the performance of the ensemble models in the two languages. While the stem, lemma, and WordPiece

TABLE 24. Review examples with incorrect predictions.

Language	Review	Gold	Predicted
English	Nice ambience, but highly overrated place.	Positive	Negative
English	The decor is rustic, traditional Japanese.	Neutral	Positive
English	Furthermore, while the fish is unquestionably fresh, rolls tend to be inexplicably bland.	Positive	Negative
English	I don't know if I'll be back....	Negative	Positive
Turkish	Bu konuda onların eline kolay kolay kimse su süremez. (correct writing: su dökemez) (No one can do better than them easily on this matter.)	Positive	Negative
Turkish	Bu fiyata böyle bir memnuniyet çok az rastlanacak şey. (Such a satisfaction at this price is rarely encountered.)	Positive	Negative
Turkish	Dışarıdan lüks gibi görünse de artık tarihi geçmiş mekan görüntüsü vermekte ve içerideki aksesuarlar göz yormakta. (Although it looks luxurious from the outside, it now has an outdated appearance and the accessories inside are visually tiring.)	Negative	Positive
Turkish	İş fiyatlara gelince öyle bir mekan için normal fakat beyti kalitesinde Avrupa veya Amerika'da yer iseniz 3 kat fazlasını ödemek zorundasınız. (When it comes to prices, it is normal for such a place, but you have to pay three times more if you eat with beyti quality in Europe or America.)	Neutral	Negative

TABLE 25. Accuracy results for ensemble models. The value in parenthesis in the last column denotes the increase over the best result for the dataset.

Dataset	Word2Vec+ GloVe	BERT	RoBERTa / XLM-RoBERTa	Acc.
IMDB Movie Reviews	Words-Pre (CNN-Dyn) Stem (CNN-Dyn) Lemma (CNN-Dyn)	Words (FFNN) Words (CNN) WordPiece (FFNN) WordPiece (CNN)	Words (FFNN) Words (CNN) WordPiece (FFNN) WordPiece (CNN)	**94.71** (1.40)
Semeval 2016 Restaurant Reviews	Words-Pre (CNN-Dyn) Stem (CNN-Dyn) WordPiece (CNN-Dyn)	Words (FFNN) Words (CNN) WordPiece (FFNN) WordPiece (CNN) Stem (FFNN) Stem (CNN)	Words (FFNN) Words (CNN)	**88.46** (2.02)
Semeval 2017 Twitter	—	Words (FFNN) Words (CNN) WordPiece (FFNN) WordPiece (CNN) Lemma (FFNN) Lemma (CNN)	Words (FFNN) Words (CNN) WordPiece (FFNN) WordPiece (CNN) Lemma (FFNN) Lemma (CNN)	**72.10** (1.44)
Semeval 2016 Turkish Restaurant Reviews	—	WordPiece (CNN)	Words (FFNN) Words-Pre (FFNN) Words-Pre (CNN)	**87.16** (2.43)
BeyazPerde Movie Reviews	Words (CNN-Dyn) Words-Pre (CNN-Dyn)	Words (FFNN) Words (CNN) WordPiece (FFNN) WordPiece (CNN)	Words (FFNN) Words (CNN) Words-Pre (FFNN) Words-Pre (CNN) ULM (FFNN) ULM (CNN)	**95.22** (0.83)

TABLE 26. Accuracy results for the best unique ensemble model. The value in parenthesis in the last column denotes the increase over the best result for the dataset.

Ensemble	Dataset	Accuracy
Words-BERT (FFNN) Words-BERT (CNN) Words-RoBERTa (FFNN) Words-RoBERTa (CNN) WordPiece-BERT (FFNN) WordPiece-BERT (CNN) WordPiece-RoBERTa (FFNN) WordPiece-RoBERTa (CNN) Lemma-BERT (FFNN) Lemma-BERT (CNN) Lemma-RoBERTa (FFNN) Lemma-RoBERTa (CNN)	IMDB Movie Reviews	**94.68** (1.37)
	Semeval 2016 Restaurant	**87.72** (1.28)
	Semeval 2017 Twitter	**72.10** (1.44)
	Semeval 2016 Turk. Res.	**85.81** (1.08)
	BeyazPerde Movie	**95.11** (0.72)

tokenizations increase the success rates in English, the WordPiece and ULM tokenizations yield high-performance results in Turkish.

In order to arrive at a general ensemble model that gives high success rates for both languages and all datasets, we make an additional experiment by combining the tokenization methods and embeddings that showed high performance independent of the language and the dataset. As stated in Section VI, the models based on contextualized embeddings outperform the models based on non-contextualized embeddings by a large margin. We thus take the most successful three tokenization methods with BERT and RoBERTa/XLMRoBERTa embeddings, which are the words, WordPiece, and lemma tokenizations for both FFNN and CNN models. We ensemble these tokenization methods for each dataset. Table 26 shows the accuracy results and the increase over the best outcome for each dataset given in Tables 9-21. We see that the ensemble model increases the performance of the individual models for all the datasets and can be regarded as a general model that works for the languages and the datasets being concerned. As a final comment, comparing the results in Table 26 with those in Table 25 signals a slight decrease in the accuracy values. This is an expected result since we use the best ensemble model specific to each dataset in Table 25, but a general ensemble model in Table 26.

VIII. DISCUSSION

In this section, we give a brief discussion on the outcomes of the experiments, the practical implications of the results, and the limitations of the work.

We observed that BERT-, RoBERTa-, and XLM-RoBERTa-based models outperform Word2Vec/GloVe-based models by a large margin. This may be regarded as an expected result showing the power of contextualized embeddings and the transformer-based encoder architecture of the BERT models. Also, RoBERTa offers higher performance than BERT in the Twitter dataset for all tokenization methods by a large margin. This may be attributed to RoBERTa embeddings being more robust on informal and noisy texts.

The analysis of the tokenization methods showed that the method used has a significant effect on sentiment classification performance. The Word2Vec- and GloVe-based models work well with the preprocessed forms or the stem forms of words, while BERT-based models get high accuracies with raw words, WordPiece, and lemma forms. We did not observe a significant difference between the two languages in terms of the success of the models and the tokenization methods. For the morphologically rich Turkish language, we would expect the more sophisticated tokenization methods based on morphemes, syllables, and partial surface forms to contribute to the performance. However, we could not observe an increase in the accuracy in most of the cases. We derived ensemble models for the datasets by combining the most successful tokenization methods in each dataset. We also formed a general ensemble model that shows high performance for both languages and all the datasets.

We conducted the experiments using the original train and test splits of the datasets to compare the results we obtained with the results in the literature. However, to avoid overfitting to the splits and to generalize the results, we repeated the experiments with k-fold cross-validation. Both sets of experiments mostly yielded similar results.

As the practical implications of the results, we can differentiate between the two cases. When non-contextualized word embeddings (Word2Vec, GloVe, etc.) are used for sentiment analysis, using the preprocessed forms of the words for both English and Turkish yields high success rates. In addition, the stem forms in English and the syllables in Turkish give a comparable performance. When contextualized word embeddings (BERT, RoBERTa, etc.) are used, using the words in surface form or preprocessed form and using WordPiece achieve the best results for both languages. These results signal that while non-contextualized embeddings necessitate more complex processing on the words, it mostly suffices using the bare forms of the words in the case of contextualized embeddings. Another observation that may help in practice is that using different tokenization methods and models and combining them with an ensembling approach increases the performance by around 1-3%.

We finally touch on a few potential limitations of this work below, addressing of which in future research may enhance the overall quality and the robustness of the findings.

- Since the datasets we used are not equally distributed over classes, it can lead to biased learning and inaccurate predictions, especially for minority classes. This can be handled by employing techniques such as oversampling or undersampling to balance the class distribution or by using more advanced methods like synthetic data generation or cost-sensitive learning algorithms.
- The work is restricted to the variants of the BERT model in generating contextualized embeddings and obtaining class decisions. Other pretrained language models and embedding methods can also be employed.
- The work covers two languages, one from the analytic language family (English) and the other from the agglutinative language family (Turkish). The proposed approaches can also be tested on languages from other families to see the generalizability of the results.

IX. CONCLUSION

In this work, we analyzed the effects of different neural models, embedding types, and tokenization methods on the performance of sentiment classification for both English and Turkish languages. We used a feed-forward neural network and a convolutional neural network as the classification model. We fed these models with two types of embeddings, a combination of non-contextualized embeddings Word2Vec and GloVe and contextualized embeddings BERT and RoBERTa (XLM-RoBERTa).

We also made a comprehensive analysis of tokenization methods. In addition to the commonly-used subword methods (BPE, WordPiece, ULM), we tested the use of raw words, different forms of words (preprocessed, raw words with stopwords eliminated, lemma, and stem forms), morphemes, and (for Turkish) syllables and partial surface forms. The experiments were conducted on three English and two Turkish datasets, which are widely used as benchmark datasets in sentiment analysis. The results showed that the tokenization methods and the embedding types highly affect the performance of the models.

For both languages, we used the pretrained BERT-based models, which employ WordPiece and BPE tokenizations. We could not pretrain these models due to the scarcity of computational resources. As future work, these models can be pretrained with all the other tokenization methods to improve the results. The language-specific tokenization methods may also be used in other NLP tasks. Another future direction may be exploring methods to enhance the interpretability of the models, such as attention visualization, feature importance analysis, or utilizing explainable AI techniques, which can provide insights into how the models arrive at their predictions.

REFERENCES

[1] S. Raman, V. Gupta, P. Nagrath, and K. Santosh, "Hate and aggression analysis in NLP with explainable AI," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 36, no. 15, Dec. 2022, Art. no. 2259036.

[2] J. Rana, L. Gaur, and K. Santosh, "Classifying customers' journey from online reviews of Amazon fresh via sentiment analysis and topic modelling," in *Proc. 3rd Int. Conf. Comput., Autom. Knowl. Manage. (ICCAKM)*, Dubai, United Arab Emirates, Nov. 2022, pp. 1–6.

[3] C. Sammut and G. I. Webb, "TF–IDF," in *Encyclopedia of Machine Learning*. Cham, Switzerland: Springer, 2010, pp. 986–987.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–12.

[5] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[8] T. Fujioka, D. Bertero, T. Homma, and K. Nagamatsu, "Addressing ambiguity of emotion labels through meta-learning," 2019, *arXiv:1911.02216*.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.

[10] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," 2015, *arXiv:1508.07909*.

[11] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 5149–5152.

[12] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 66–75.

[13] F. Katamba and J. Stonham, "Morphology," in *Macmillan Modern Linguistics*. U.K.: Macmillan Education, 2006.

[14] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[15] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd ed. London, U.K.: Pearson, 2019.

[16] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[17] Y. Wu et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[18] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol. (ACL)*, 2011, pp. 142–150.

[19] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiit, "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (ACL)*, 2016, pp. 19–30.

[20] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 502–518.

[21] A. Ucan, B. Naderalvojoud, E. A. Sezer, and H. Sever, "SentiWordNet for new language: Automatic translation approach," in *Proc. 12th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov. 2016, pp. 308–315.

[22] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.Zip: Compressing text classification models," 2016, *arXiv:1612.03651*.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. (2019). *Language Models are Unsupervised Multitask Learners*. [Online]. Available: https://api.semanticscholar.org/CorpusID:160025533

[24] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Sys.*, vol. 33, 2020, pp. 1877–1901.

[25] Y. Goldberg and O. Levy, "Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*.

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[27] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave, "CCNet: Extracting high quality monolingual datasets from web crawl data," 2019, *arXiv:1911.00359*.

[28] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*. New York, NY, USA: O'Reilly Media, 2009.

[29] M. F. Porter, "An algorithm for suffix stripping," in *Readings in Information Retrieval*. Burlington, MA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 313–316.

[30] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.

[31] H. Sak, T. Gngr, and M. Saralar, "Turkish language resources: Morphological parser, morphological disambiguator, and web corpus," in *Advances in Natural Language Processing*. Cham, Switzerland: Springer, 2008, pp. 417–427.

[32] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vols. 11–38, pp. 39–41, Jan. 1995.

[33] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A Python natural language processing toolkit for many human languages," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2020, pp. 1–8.

[34] A. Carstairs-McCarthy, *An Introduction to English Morphology*. Edinburgh, U.K.: Edinburgh Univ. Press, 2002.

[35] S. M. Oztaner, "A word grammar of Turkish with morphophonemic rules," 1996, *arXiv:cmp-lg/9608013*.

[36] S. Virpioja, P. Smit, S.-A. Grnroos, and M. Kurimo, "Professor 2.0: Python implementation and extensions for morfessor baseline," Aalto Univ., Espoo, Finland, Tech. Rep. 25, 2013.

[37] P. Gage, "A new algorithm for data compression," *C Users J.*, vol. 12, p. 2338, Jan. 1994.

[38] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2018, pp. 66–71.

[39] İlknur Büyükkuşcu ve Eşref Adalı, "Heceleme Yöntemiyle Kök Sözcük Üretme," *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 2, no. 1, 2016. [Online]. Available: https://dergipark.org.tr/tr/pub/tbbmd/issue/22241/238761#article_cite

[40] C. R. Aydın and T. Güngör, "Sentiment analysis in Turkish: Supervised, semi-supervised, and unsupervised techniques," *Natural Lang. Eng.*, vol. 27, no. 4, pp. 455–483, Jul. 2021.

[41] S. Suneetha and S. V. Row, "Aspect-based sentiment analysis: A comprehensive survey of techniques and applications," *J. Data Acquisition Process.*, vol. 38, no. 3, pp. 177–203, 2023.

[42] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1997.

[43] A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, "k-nearest neighbor classification," in *Data Mining in Agriculture*. New York, NY, USA: Springer-Verlag, 2009, pp. 83–106.

[44] C. Sutton, K. Rohanimanesh, and A. McCallum, "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 282–289.

[45] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[46] G. I. Webb, "Naïve Bayes," *Encyclopedia Mach. Learn.*, vol. 15, no. 1, pp. 713–714, 2010.

[47] N. Kamal, "Using maximum entropy for text classification," in *Proc. IJCAI Workshop Mach. Learn. Inf. Filtering*, 1999, pp. 61–67.

[48] Z. Nasim, Q. Rajput, and S. Haider, "Sentiment analysis of student feedback using machine learning and lexicon based approaches," in *Proc. Int. Conf. Res. Innov. Inf. Syst. (ICRIIS)*, Jul. 2017, pp. 1–6.

[49] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, Mar. 2020.

[50] Z. Kastrati, L. Ahmedi, A. Kurti, F. Kadriu, D. Murtezaj, and F. Gashi, "A deep learning sentiment analyser for social media comments in low-resource languages," *Electronics*, vol. 10, no. 10, p. 1133, May 2021.

[51] R. Chandra and A. Krishna, "COVID-19 sentiment analysis via deep learning during the rise of novel cases," *PLoS ONE*, vol. 16, no. 8, Aug. 2021, Art. no. e0255615.

[52] B. T. Hung, "Domain-specific vs. general-purpose word representations in sentiment analysis for deep learning models," in *Frontiers in Intelligent Computing: Theory and Applications*. Singapore: Springer, 2020, pp. 252–264.

[53] B. Shin, T. Lee, and J. D. Choi, "Lexicon integrated CNN models with attention for sentiment analysis," 2016, *arXiv:1610.06272*.

[54] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," 2017, *arXiv:1707.01780*.

[55] L. Haonan, S. H. Huang, T. Ye, and G. Xiuyan, "Graph star net for generalized multi-task learning," 2019, *arXiv:1906.12330*.

[56] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, "Entailment as few-shot learner," 2021, *arXiv:2104.14690*.

[57] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" 2019, *arXiv:1905.05583*.

[58] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," 2019, *arXiv:1904.12848*.

[59] T. Khalil and S. R. El-Beltagy, "NileTMRG at SemEval-2016 task 5: Deep convolutional neural networks for aspect category and sentiment extraction," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 271–276.

[60] A. Kumar, S. Kohail, A. Kumar, A. Ekbal, and C. Biemann, "IIT-TUDA at SemEval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 1129–1135.

[61] C. Brun, J. Perez, and C. Roux, "XRCE at SemEval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 277–281.

[62] S. Aït-Mokhtar, J.-P. Chanod, and C. Roux, "Robustness beyond shallowness: Incremental deep parsing," *Natural Lang. Eng.*, vol. 8, nos. 2–3, pp. 121–144, Jun. 2002.

[63] P. C. Hansen, "The truncated SVD as a method for regularization," Stanford Univ., Stanford, CA, USA, Tech. Rep. NA-86-36, 1986.

[64] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

[65] O. Wallaart and F. Frasincar, "A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models," in *The Semantic Web* (Lecture Notes in Computer Science), vol. 11503. Cham, Switzerland: Springer, 2019, pp. 363–378.

[66] S. Zheng and R. Xia, "Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention," 2018, *arXiv:1802.00892*.

[67] M. M. Trusca, D. Wassenberg, F. Frasincar, and R. Dekker, "A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention," 2020, *arXiv:2004.08673*.

[68] N. Reddy, P. Singh, and M. M. Srivastava, "Does BERT understand sentiment? Leveraging comparisons between contextual and non-contextual embeddings to improve aspect-based sentiment models," 2020, *arXiv:2011.11673*.

[69] M. Cliche, "BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs," 2017, *arXiv:1704.06125*.

[70] Y. Yin, Y. Song, and M. Zhang, "NNEMBs at SemEval-2017 task 4: Neural Twitter sentiment classification: A simple ensemble method with different embeddings," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 621–625.

[71] T. Lei, H. Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, A. Moschitti, and L. Marquez, "Semi-supervised question retrieval with gated convolutions," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1279–1289.

[72] H. Hamdan, "Senti17 at SemEval-2017 task 4: Ten convolutional neural network voters for tweet polarity classification," 2017, *arXiv:1705.02023*.

[73] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 747–754.

[74] S. Ruder, P. Ghaffari, and J. G. Breslin, "INSIGHT-1 at SemEval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 330–336.

[75] C. R. Aydın, T. Güngör, and A. Erkan, "Generating word and document embeddings for sentiment analysis," in *Proc. 20th Int. Conf. Intell. Text Process. Comput. Linguistics*, La Rochelle, France, 2019.

[76] P. D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2001, Art. no. 417424.

[77] M. Kaya, G. Fidan, and I. H. Toroslu, "Sentiment analysis of Turkish political news," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1, Dec. 2012, pp. 174–180.

[78] M. F. Amasyali, "Active learning for Turkish sentiment analysis," in *Proc. IEEE INISTA*, Jun. 2013, pp. 1–4.

[79] C. Türkmenoğlu and A. C. Tantuğ, "Sentiment analysis in Turkish media," in *Proc. Workshop Issues Sentiment Discovery Opinion Mining*, 2014, p. 111.

[80] G. Vural, B. Cambazoğlu, Ö. Z. Tokgöz, and P. Karagöz, "A framework for sentiment analysis in Turkish: Application to polarity detection of movie reviews in Turkish," in *Proc. 27th Int. Symp. Comput. Inf. Sci.*, 2012, pp. 437–445.

[81] R. Dehkharghani, Y. Saygin, B. Yanikoglu, and K. Oflazer, "SentiTurkNet: A Turkish polarity lexicon for sentiment analysis," *Lang. Resour. Eval.*, vol. 50, no. 3, pp. 667–685, Sep. 2016.

[82] Yelp, Inc. (2022). *Yelp Open Dataset*. Yelp Dataset Challenge. [Online]. Available: https://www.yelp.com/dataset

[83] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[84] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, Feb. 2023.

ALI ERKAN (Member, IEEE) received the B.S. and M.S. degrees in industrial engineering from Bilkent University, Ankara, and the M.S. degree in software engineering from Boğaziçi University, Istanbul, where he is currently pursuing the Ph.D. degree in computer engineering. He was a Software Engineer at various software development companies. Since 2022, he has been an Instructor with Koç University, Istanbul. His research interests include machine learning, natural language processing, and sentiment analysis.

TUNGA GÜNGÖR received the M.S. and Ph.D. degrees from the Department of Computer Engineering, Boğaziçi University. He was a Visiting Professor with the Center for Language and Speech Technologies and Applications, Universitat Politècnica de Catalunya, Barcelona, Spain, from 2011 to 2012. He is currently a Senior Lecturer and a Researcher with the Department of Computer Engineering, Boğaziçi University. He is a member with the Artificial Intelligent Laboratory and the Text Analytics and Bioinformatics Laboratory. He teaches undergraduate and graduate-level courses on the topics of artificial intelligence, natural language processing, machine translation, and algorithm analysis. He participated as the Project Leader in projects about developing an adaptive question answering system for primary and secondary education students, developing concept mining methods for document analysis, developing a hand-written recognition system using a large lexicon, morphology-based language modeling for speech recognition, and developing structure-preserving and query-biased automated summarization methods. The projects were funded by the Turkish Scientific and Technological Research Council of Turkey and the national funds. He has published about 90 scientific articles and participated in several research projects and conference organizations. His research interests include natural language processing, machine translation, machine learning, and pattern recognition.