ASPECT-BASED SENTIMENT ANALYSIS FOR TURKISH USING DEEP LEARNING MODEL COMBINATIONS

by

Ali Erkan

B.S., Industrial Engineering, Bilkent University, 1995

M.S., Industrial Engineering, Bilkent University, 1997

M.S., Software Engineering, Boğaziçi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2024

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Tunga Güngör. He has always guided me with great wisdom and care throughout my PhD. His continued support and understanding were one of the most fundamental aspects that have helped me to complete this journey. I am truly grateful to have worked with such a great supervisor and would like to thank him for all the effort he has put in me and this thesis.

I would like to thank my thesis committee members Prof. Arzucan Özgür and Prof. Emin Erkan Korkmaz for their valuable advices that have helped me to improve my research and come up with new ideas. Moreover, I would like to thank Prof. Fikret Gürgen and Assoc. Prof. Tuğba Dalyan for kindly accepting to be part of my thesis jury and for their valuable comments.

I would like to thank my parents and brothers for always supporting me throughout my life. They have always encouraged me to aim higher and trusted in me.

Finally, but the most importantly, I would like to thank my family and especially my dear wife Ayşe for her patience and support whom have helped me share this burden throughout all these years. This journey would have not been possible without her continued support and encouragements. My warmest thanks go to my lovely sons, Batuhan and Berkay. The thought of having the support of this wonderful family has given me the strength to always continue on this long and hard path.

ABSTRACT

ASPECT-BASED SENTIMENT ANALYSIS FOR TURKISH USING DEEP LEARNING MODEL COMBINATIONS

People use the World Wide Web heavily to share their experiences with entities such as products, services or travel destinations. These reviews and comments are essential for consumer decisions and create a valuable source that may be used to measure satisfaction. Sentiment analysis (SA) is a natural language processing task of identifying opinions expressed in such text fragments. Aspect-Based Sentiment Analysis (ABSA) is a research area whose ability to provide fine-grained insights into sentiment expressions related to specific aspects or features of entities and it has gained significant attention in recent years. In this thesis, we mainly focus on sentiment analysis and aspect extraction parts of ABSA especially in Turkish. For SA, we used two approaches. The first is to develop average review vectors from noncontextualized and contextualized embeddings and add weights to these review vectors using word frequencies in positive and negative sensitivity-tagged reviews. The second is to use different combinations of deep learning models with different embedding methods and tokenization approaches. We make a comprehensive analysis of eleven different tokenization approaches, including the commonly used subword methods and morphologically motivated segmentations. We ensemble the models to find out which tokenization approaches produce better results together. The final part of the thesis is related to ABSA's crucial aspect extraction component, which involves identifying and extracting aspect terms from reviews. We develop a novel tree positional embedding, in addition to embeddings for tokens and Part-of-Speech (POS) tags, and deep learning algorithms are used to develop a novel learning model. A novel dataset is created from English reviews by translating them into Turkish and relabelling them in addition to an existing Turkish review dataset.

ÖZET

TÜRKÇE İÇİN DERİN ÖĞRENME MODELLERİ KULLANARAK ÖZELLİK BAZLI DUYGU ANALİZİ

İnsanlar ürünler ve hizmetler gibi varlıklarla yaşadıkları deneyimlerini paylaşmak için World Wide Web'i yoğun bir şekilde kullanmaktadır. Bu incelemeler ve yorumlar tüketici kararları için önemlidir ve memnuniyeti ölçmek için kullanılabilecek değerli bir kaynak oluşturur. Duygu analizi, bu tür metin parçalarında ifade edilen düşünceleri belirlemeye yönelik bir doğal dil işleme görevidir. Özellik Tabanlı Duygu Analizi (ABSA), varlıkların belirli özelliklerine ilişkin duygu ifadelerine ayrıntılı bilgiler sağlama yeteneği olan ve son yıllarda büyük ilgi gören bir araştırma alanıdır. Bu tezde ağırlıklı olarak ABSA'nın özellikle Türkçe'de duygu analizi ve özellik çıkarma kısımlarına odaklanılmıştır. Duygu analizi için iki yaklaşım kullanılmıştır. Bunlardan ilki, bağlamsal olmayan ve bağlamsal olan ortalama gömme vektörleri geliştirmek ve pozitif ve negatif hassasiyet etiketli yorumlardaki kelime sıklıklarını kullanarak bu gömme vektörlerine ağırlıklar eklemektir. İkincisi, farklı gömme yöntemleri ve tokenizasyon yaklaşımlarıyla derin öğrenme modellerinin farklı kombinasyonlarını kullanmaktır. Yaygın olarak kullanılan alt kelime yöntemleri ve morfolojik olarak nitelenen bölümlemeler dahil olmak üzere on bir farklı tokenizasyon yaklaşımının kapsamlı bir analizi yapılmıştır. Tezin son kısmı, ABSA'nın önemli özellik çıkarma bileşeniyle ilgilidir; bu bileşen, yorumlardan özellik terimlerinin tanımlanmasını ve çıkarılmasını içerir. Belirteçler ve Konuşma Kısmı (POS) etiketleri için yerleştirmelere ek olarak yeni bir ağaç konumsal yerleştirme kullanılmıştır ve yeni bir öğrenme modeli geliştirmek için derin öğrenme algoritmaları kullanılmıştır. Mevcut Türkçe yorumların yanı sıra İngilizce yorumlar da Türkçeye çevrilerek ve yeniden etiketlenerek yeni bir veri seti oluşturulmuştur. Bulgularımız, en son teknikleri birleştirmenin etkinliğini göstererek, nesneye dayalı duygu analizinin geliştirilmesine katkıda bulunmuştur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . ix
LIST OF TABLES . . . xi
LIST OF SYMBOLS . . . xv
LIST OF ACRONYMS/ABBREVIATIONS . . . xvi
1. INTRODUCTION . . . 1
1.1. Aspect-Based Sentiment Analysis . . . 1
1.2. Sentiment Analysis/Classification . . . 2
1.3. Word Embeddings . . . 4
1.3.1. Word2Vec . . . 4
1.3.2. GloVe . . . 4
1.3.3. BERT . . . 4
1.3.4. RoBERTa . . . 5
1.3.5. XLM-RoBERTa . . . 5
1.4. Tokenizations . . . 5
1.4.1. Words . . . 6
1.4.2. Words - Preprocessed . . . 6
1.4.3. Words - No Stopwords . . . 6
1.4.4. Stem . . . 6
1.4.5. Lemma . . . 7
1.4.6. Morphemes . . . 7
1.4.7. Byte Pair Encoding . . . 7
1.4.8. WordPiece . . . 8
1.4.9. Unigram Language Model . . . 8
1.4.10. Syllables . . . 8
1.4.11. Partial Surface Form . . . 9

1.5. Aspect Extraction . . . 11
1.6. Contributions . . . 12
1.6.1. Sentiment Analysis Using Averaged Weighted Word Vector Features . . . 12
1.6.2. Analysis of Deep Learning Model Combinations and Tokenization Approaches in Sentiment Classification . . . 12
1.6.3. Aspect Extraction from Turkish Reviews by using BERT, BiLSTM, CRF, and Tree Positional Encoding . . . 13
1.7. Thesis Outline . . . 13
2. RELATED WORK . . . 14
2.1. Sentiment Analysis . . . 14
2.2. Aspect Extraction . . . 20
3. DATASETS . . . 23
4. SENTIMENT ANALYSIS USING AVERAGED WEIGHTED WORD VECTOR FEATURES . . . 27
4.1. Model . . . 27
4.1.1. Averaged Review Vector (ARV) and Weighted Averaged Review Vector (WARV) . . . 29
4.1.2. Convolutional Neural Network (CNN) . . . 32
4.1.3. Ensemble . . . 34
4.2. Experiments . . . 36
4.2.1. IMDB Reviews . . . 37
4.2.1.1. McNemar's Test . . . 37
4.2.1.2. Ensemble . . . 38
4.2.2. Semeval 2016 Task 5 Dataset . . . 39
4.2.3. Yelp Dataset . . . 43
4.3. Conclusion . . . 44
5. DEEP LEARNING MODEL COMBINATIONS AND TOKENIZATION APPROACHES IN SENTIMENT CLASSIFICATION . . . 46
5.1. Models . . . 48
5.1.1. CNN With Word2Vec And GloVe Embeddings . . . 48

5.1.2. BERT/RoBERTa/XLM-RoBERTa And FFNN 49
5.1.3. CNN With BERT/RoBERTa/XLM-RoBERTa Embeddings . . . 49
5.2. Experiments 51
5.3. Ensemble 68
5.4. Discussion 72
5.5. Conclusion 74
6. ASPECT EXTRACTION FROM TURKISH REVIEWS BY USING BERT, BILSTM, CRF, AND TREE POSITIONAL ENCODING 75
6.1. Model 75
6.1.1. Word Embeddings 76
6.1.2. POS Tag Embeddings 77
6.1.3. Position Encoding 77
6.1.4. BiLSTM And CRF Layers 80
6.1.5. Viterbi Decoding 81
6.1.6. Evaluation 81
6.2. Experiments And Results 82
6.3. Sentiment Analysis 86
6.4. Conclusion 87
7. CONCLUSION 89
7.1. Summary of Contributions 89
7.2. Future Work 91
REFERENCES 92
APPENDIX A: TELİF HAKLARI 112
APPENDIX B: IEEE LICENSES AND COPYRIGHT 113
APPENDIX C: PLOS ONE LICENSES AND COPYRIGHT 114

LIST OF FIGURES

Figure 1.1. Syllable extraction algorithm for Turkish. 9

Figure 4.1. Our feed-forward neural network model. 31

Figure 4.2. CNN sentence classification. 34

Figure 4.3. Ensemble of learning algorithms. 35

Figure 5.1. CNN model with Word2Vec and GloVe embeddings. 50

Figure 5.2. FFNN model with Bert/RoBERTa/XLM-RoBERTa embeddings. 50

Figure 5.3. CNN model with Bert/RoBERTa/XLM-RoBERTa embeddings. . 50

Figure 5.4. Accuracy results for Stanford IMDB movie reviews dataset. . . . 52

Figure 5.5. Macro F1-values for Semeval 2016 restaurant reviews dataset. . . 56

Figure 5.6. Macro F1-values for Semeval 2017 Twitter dataset. 58

Figure 5.7. Macro F1-values for Semeval 2016 Turkish restaurant reviews dataset. 61

Figure 5.8. Accuracy results for BeyazPerde movie reviews dataset. 62

Figure 6.1. (BERT-)BiLSTM-CRF model 76

Figure 6.2. Dependency parse tree of the sentence ”son olarak dün gittiğim her gittiğimde ayrı keyif aldığım güzel mekan .”. 79

Figure 6.3. Sentiment classification for extracted aspects. 87

LIST OF TABLES

Table 1.1. Aspects of some Turkish customers' restaurant reviews. 2

Table 1.2. Tokenizations of example sentences. 10

Table 2.1. Summary of related works on IMDB movie reviews dataset. 16

Table 2.2. Summary of related works on semeval 2016 restaurant reviews dataset. 17

Table 2.3. Summary of related works on Semeval 2017 Twitter dataset. . . . 18

Table 2.4. Summary of related works on Semeval 2016 Turkish restaurant reviews dataset. 18

Table 2.5. Summary of related works on Beyazperde movie reviews dataset. . 19

Table 2.6. List of embedding methods/features and the learning models used in related studies. 19

Table 3.1. Number of reviews in the datasets. 23

Table 3.2. Number of reviews in the datasets. 24

Table 3.3. Aspects of English and translated(Turkish) reviews. 24

Table 4.1. Reviews with ids (M=6). 31

Table 4.2. IMDB reviews with ids. 33

Table 4.3. Accuracies of methods before ensemble for IMDB dataset. 36

Table 4.4. McNemar's test results of the methods. 38

Table 4.5. Accuracies of methods after ensemble for IMDB dataset. 40

Table 4.6. Accuracies of methods (Ensemble-2) for Semeval restaurant dataset. 41

Table 4.7. Accuracies for Yelp restaurant dataset with 100.000 reviews. 43

Table 5.1. Accuracy results for Stanford IMDB movie reviews dataset. 51

Table 5.2. K-fold accuracy results for Stanford IMDB movie reviews dataset. 53

Table 5.3. Accuracy results for Semeval 2016 restaurant reviews dataset. . . 54

Table 5.4. Macro F1-values for Semeval 2016 restaurant reviews dataset. . . . 54

Table 5.5. K-fold macro F1-values for Semeval 2016 restaurant reviews dataset. 55

Table 5.6. Accuracy results for Semeval 2017 Twitter dataset. 56

Table 5.7. Macro F1-values for Semeval 2017 Twitter dataset. 57

Table 5.8. K-fold macro F-1 values for Semeval 2017 Twitter dataset. 58

Table 5.9. Accuracy results for Semeval 2016 Turkish restaurant reviews dataset. 59

Table 5.10. Macro F1-values for Semeval 2016 Turkish restaurant reviews dataset. 60

Table 5.11. K-fold macro F-1 values for Semeval 2016 Turkish restaurant reviews dataset. 61

Table 5.12. Accuracy results for BeyazPerde movie reviews dataset. 62

Table 5.13. K-fold accuracy results for BeyazPerde movie reviews dataset. . . 63

Table 5.14. Stability analysis of models over tokenization methods, the maximum and average accuracies and the standard deviation are shown for each model and dataset. 65

Table 5.15. Stability analysis of tokenization methods over models, the standard deviation averaged over the models is shown for each model and English/Turkish datasets. 66

Table 5.16. Review examples with incorrect predictions. 67

Table 5.17. Accuracy results for ensemble models, the value in parenthesis in the last column denotes the increase over the best result for the dataset. 70

Table 5.18. 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. 71

Table 6.1. BiLSTM-CRF model macro F1-values for Turkish SemEval 2016 restaurant reviews dataset. 83

Table 6.2. BERT-BiLSTM-CRF model macro F1-values for Turkish SemEval 2016 restaurant reviews dataset. 83

Table 6.3. BiLSTM-CRF model macro F1-values for translated English SemEval 2016 restaurant reviews dataset. 84

Table 6.4. BERT-BiLSTM-CRF model macro F1-values for translated English SemEval 2016 restaurant reviews dataset. 85

Table 6.5. Comparison of models for Turkish SemEval 2016 restaurant reviews dataset. 86

Table 6.6. Comparison of models for English SemEval 2016 restaurant reviews dataset. 86

Table 6.7. Sentiment classification for extracted aspects. 87

LIST OF SYMBOLS

b_i	Weights for word vector of i^{th} word
d	Dimension of position vector
d_i	Dimension of vector i
$F1$	Harmonic mean of the precision and recall scores
FN	Number of false negative
FP	Number of false positive
$M^{\frac{2i}{d}}$	user-defined constant (10,000) power of 2 times word position divided by dimension
N	Number of words in sentence
P	Probability
$p_{pos,2i}$	Positional encoding of position for ith token
pos	Position in sequence
TP	Number of true positive
$\mathbf{wv}_i$	Word vector of i^{th} word
$\hat{\mathbf{wv}}_i$	Normalized vector of word vector of i^{th} word
$\lvert\mathbf{wv}_i\rvert$	Dot product of word vector of i^{th} word
wv_{ij}	j^{th} value of i^{th} word of word vector

LIST OF ACRONYMS/ABBREVIATIONS

ABSA	Aspect-Based Sentiment Analysis
AdaDelta	Name of a stochastic optimization technique
API	Application Programming Interface
ARV	Averaged Review Vector
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
BPE	Byte Pair Encoding
BOUN	Boğaziçi University
CBOW	Continuous Bag of Words
CLS	Added token at the beginning of a sentence for BERT
CNN	Conditional Neural Network
CRF	Conditional Random Field
DNN	Deep Neural Network
ELMo	Embeddings from Language Model
FastText	A learning model of word embeddings and text classification
FFNN	Feed-Forward Neural Network
GPT	Generative Pre-training Transformer
GloVe	Global Vectors for distributed word representation
HMM	Hidden Markov Model
IMDB	Internet Movie DataBase
JSON	JavaScript Object Notation
k-fold	Cross validation procedure
k-NN	k-Nearest Neighbors
LLAMA	Large Language Model
LSTM	Long Short-Term Memory
NB	Naive Bayes
NBSVM	Naive Bayes-Support Vector Machine
NLP	Natural Language Processing

NLTK	Natural Language Toolkit
NN	Neural Network
PMI	Pointwise Mutual Information
POS	Part of speech
PV	Paragraph Vectors
RCNN	Region-based Convolutional Neural Network
RELU	Rectified Linear Unit
RNN	Recurrent Neural Network
RNNLM	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Approach
SEMEVAL	International Workshop on Semantic Evaluation
SentiTurkNet	Turkish Polarity Lexicon for Sentiment Analysis
SG	Skip Gram
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
TPE	Tree Positional Encoding
UD	Universal Depedencies
UDA	Unsupervised Data Augmentation
ULM	Unigram Language Model
WARV	Weighted Averaged Review Vector
WordPiece	a subword segmentation algorithm
Word2Vec	Embedding vector for words
WVM	Word Vector Model
XLM-RoBERTa	Multilingual Version of Robustly Optimized BERT Pretraining Approach
YELP	A dataset is a subset of our businesses, reviews, and user data

1. INTRODUCTION

1.1. Aspect-Based Sentiment Analysis

Aspect-based sentiment analysis (ABSA) is a subfield of natural language processing (NLP) that aims to identify the sentiment expressed towards specific aspects or features of a product, service, or entity. The task involves analyzing the text to identify the various aspects or attributes being discussed and determining the sentiment expressed towards each aspect. ABSA is an essential area of research as it has many practical applications in industries such as e-commerce, social media monitoring, and customer service. ABSA is a rapidly evolving field with new techniques and approaches being proposed regularly. As businesses continue to place increasing importance on customer feedback and sentiment analysis, ABSA is likely to play an increasingly important role in the e-commerce and customer service industries.

ABSA is a featured subfield of sentiment analysis that has received significant attention in recent years [1, 2]. Traditional sentiment analysis focuses on determining the overall polarity of a piece of text, classifying it as positive, negative, or neutral. However, in many real-world scenarios, this global sentiment classification does not provide a comprehensive view of the different kinds of opinions expressed in the text.

ABSA addresses this limitation by offering a more granular approach to sentiment analysis. Instead of treating the entire text as a single unit, ABSA aims to analyze sentiments concerning specific aspects or features of the discussed entities. Aspects can refer to particular attributes, components, or elements associated with entities such as products, services, or experiences. In a product review, for instance, ABSA enables us to observe the sentiments expressed about individual features of the product, such as its design, performance, battery life, and customer service. This fine-grained analysis provides companies and researchers with more detailed information and reflects the nuanced nature of human opinions.

Table 1.1 shows some examples of customer restaurant reviews and the aspects of the reviews.

Table 1.1. Aspects of some Turkish customers' restaurant reviews.

Reviews	**Aspects**	**Sentiment**
Gördüğüm en güzel dekorasyonlu simit sarayı diyebilirim.	dekorasyonlu	Positive
İyi bir yer belki arada denenebilir.	yer	Positive
Lokumu tavsiye ederim.	lokumu	Positive
Kahvaltı tabağının 12.5 tl den 18.5 ye çıkması gerçekten çok komik.	kahvaltı tabağının	Negative
Tüm hardallı balığı denedim ardından elmalı yeşil çay ikisi de enfesti.	hardallı balığı elmalı yeşil çay	Positive Positive

1.2. Sentiment Analysis/Classification

Sentiment analysis/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 is a subfield of the ABSA. 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. [3] analyzed the performance of different algorithms in detecting aggression and hate speech from Twitter and Facebook messages. Similarly, Rana et al. [4] 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.
- 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) [5] 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 [6], GloVe [7], and BERT [8].

1.3. 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 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 [6], GloVe [7], FastText [9], BERT [8], RoBERTa [10], XLM-RoBERTa [11], GPT-2 [12], and GPT-3 [13]. We briefly explain below the models that are used in this work.

1.3.1. Word2Vec

Word2Vec is an unsupervised embedding model developed by Mikolov et al. [6]. It has two versions: Skip Gram (SG) and Continuous Bag of Words (CBOW) [14]. The Skip Gram model with negative sampling usually yields more efficient results [15].

1.3.2. 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.

1.3.3. 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.

1.3.4. 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. [10] used a new large dataset.

1.3.5. 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 [16].

1.4. 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.

1.4.1. 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.

1.4.2. 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.

1.4.3. 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 [17].

1.4.4. 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 [18].

In this approach, we use the stems of words as tokens in the sentiment analysis models. For stemming, we use Stanford CoreNLP [19] for English and the morphological parser of Sak et al. [20] for Turkish. Also, we used Sak et al.'s morphological parser in lemmatization and morpheme extraction for Turkish.

1.4.5. Lemma

Lemmatization [21] 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 [22] for the lemmatization in English.

1.4.6. 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 [23] [24]. We use Morfessor [25] for English morpheme extraction.

1.4.7. Byte Pair Encoding

Byte pair encoding (BPE) [26] has originated from a data compression algorithm [27]. Its variants are used in Google's SentencePiece tokenization method [28] and OpenAI's GPT-3 model [13]. 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.

1.4.8. WordPiece

The WordPiece algorithm [29] 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.

1.4.9. 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) [30] 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.

1.4.10. 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 1.1 [31]. 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.

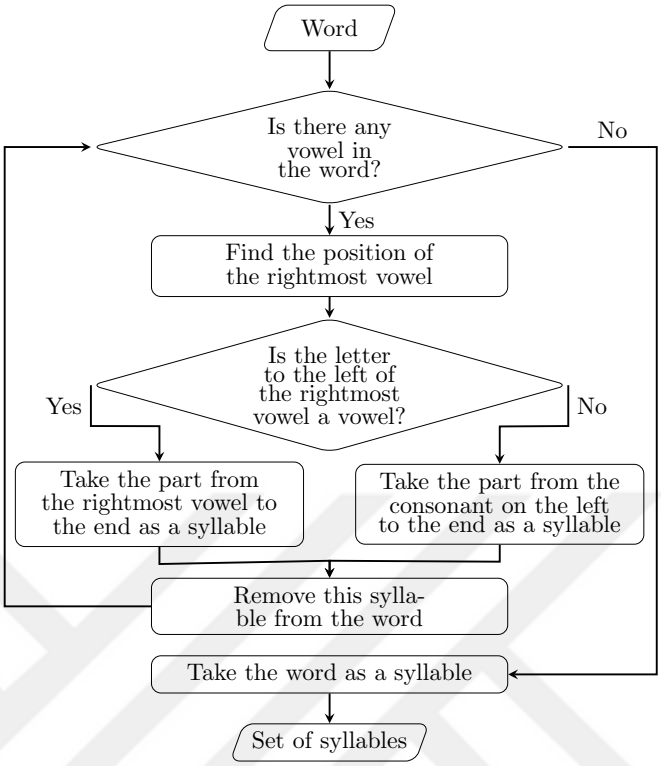


Figure 1.1. Syllable extraction algorithm for Turkish.

1.4.11. Partial Surface Form

The partial surface form of a word is an intermediate form between the root form and the surface form [32]. 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.

Table 1.2. 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

Table 1.2 exemplifies the tokenization methods explained in this section. The first nine methods are shown on a sentence from the IMDB Movie Reviews and the last two are specific to Turkish on a sentence from the BeyazPerde Movie Reviews.

1.5. Aspect Extraction

The fundamental challenge in ABSA lies in accurate aspect extraction, which aims to identify and extract the aspect terms present in the text. This step forms the basis for subsequent sentiment analysis, as sentiments are linked to specific aspects. The extraction of aspects involves identifying keywords, phrases, or syntactic patterns that refer to the aspects under consideration.

In this work, we focus on aspect extraction, which is a pivotal task in understanding the fine-grained nuances of sentiments. Using the synergy of cutting-edge techniques, we propose a comprehensive model that integrates BERT (Bidirectional Encoder Representations from Transformers) [8], BiLSTM (Bidirectional Long Short-Term Memory) [33], CRF (Conditional Random Fields) [34], tree positional encoding (TPE), and different types of embeddings for enhanced aspect extraction. By integrating different types of models, we aim to handle the different requirements in aspect extraction.

BERT, renowned for its contextualized word representations, allows the model to capture intricate semantic relationships within the text. BiLSTM, with its ability to model long-term dependencies, addresses the hierarchical and sequential nature of aspects, which is essential for accurate extraction. Incorporating CRF as a final layer augments the model's performance by considering the sequential dependencies between the terms and thus imposing constraints between adjacent aspect predictions. Furthermore, exploring pre-trained word embeddings and tree positional encoding (TPE) in conjunction with the neural network architecture enhances the model's sensitivity to linguistic subtleties, contributing to a more nuanced understanding of aspects.

1.6. Contributions

The main contributions of this thesis can be summarized as follows:

1.6.1. Sentiment Analysis Using Averaged Weighted Word Vector Features

We develop two methods that combine different types of word vectors to learn and estimate the polarity of reviews. We create average review vectors from word vectors and add weights to these review vectors using word frequencies in positive and negative sensitivity-tagged reviews. We applied the methods to several datasets from different domains used as standard sentiment analysis benchmarks. We ensemble the techniques with each other and existing methods, and compare them with the approaches in the literature. The results show that the performances of our approaches outperform the state-of-the-art success rates. Parts of this work have been presented at the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CiCLing 2019) and published in the *Springer Lecture Notes in Computer Science* [35]. This work has been published in *Plos ONE* [36].

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

We analyze the effects of different combinations of deep learning models, 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. This work has been published in *IEEE Access* [37].

1.6.3. Aspect Extraction from Turkish Reviews by using BERT, BiLSTM, CRF, and Tree Positional Encoding

A novel tree positional encoding (TPE) is used in addition to pretraining of Deep Bidirectional Transformers (BERT) and Word2Vec embeddings for tokens and Part-of-Speech (POS) tags, respectively, and also Bidirectional Long Short-Term Memory (BiLSTM) and Conditional Random Fields (CRF) learning algorithms are used to develop a novel learning model. We obtain state-of-the-art results for extracting aspects from Turkish review datasets. A novel dataset is created from English reviews in addition to a Turkish review by translating them into Turkish and relabeling them. We release and make publicly available [1] a new aspect-based sentiment analysis dataset for Turkish. Our findings contribute to advancing aspect-based sentiment analysis by showcasing the effectiveness of combining state-of-the-art techniques. This work has been submitted in *Expert Systems With Applications* [38].

1.7. Thesis Outline

This thesis is outlined as follows. The relevant work on aspect-based sentiment analysis is provided in Chapter 2. Chapter 3 summarizes the datasets used in this work. Chapter 4 introduces weighted average embedding vectors to learn sentiment classification with Convolutional and Feed-Forward Neural Networks. The variety of embedding methods over different tokenizations to learn sentiment is explained in Chapter 5. In Chapter 6, we introduce the models to learn aspect extraction. Finally, Chapter 7 discusses the outcome of the work presented in this thesis to aid future research.

[1] https://github.com/alierkan/Turkish-ABSA

2. RELATED WORK

2.1. Sentiment Analysis

Texts that provide online reviews, are essential for consumer decision making [39], and customer comments create valuable sources for companies to improve customer satisfaction. Sentiment Analysis (SA) includes every aspect of Natural Language Processing, such as entity recognition, coreference resolution, and negation handling (Liu [40] and Cambria et al. [41]) In recent years, a significant amount of research has been done on aspect-based sentiment analysis, and several different approaches and methods have been proposed. In this section, we review some of the key research papers and developments in this domain.

The several workshops and conferences focus on shared tasks related to SA. Some of them are NTCIR [42]; TAC2013 [43]; SemEval-2013 Task 2 [44]; SemEval-2014 Task 4 [45]; SemEval-2015 Task 12 [46]; SemEval-2016 Task 6 [47]; Semeval-2016 Task 5 [48] ;SemWebEval 2014 [49]; GESTALT-2014 [50]; SentiRuEval [51]. These competitions provide training datasets. Currently, most of the available SA-related datasets (Socher et al. [52]; Ganu et al. [53]) are monolingual and generally focus on English texts. Furthermore, some of them provide multilingual datasets (Klinger and Cimiano [54]; Jiménez-Zafra et al. [55]) that are useful to enable the development and testing of cross-lingual methods [56]. In addition to these, there are some other datasets, such as Stanford IMDB Reviews [57] or Yelp dataset [58], and some SA studies have been carried out using these datasets, such as ([59], [60]).

The approaches used in sentiment analysis can be divided into three: rule-based, machine learning-based, and hybrid approaches. Early ABSA studies employed rule-based methods for aspect extraction. Various techniques and resources such as pattern matching, syntactic parsing, and domain-specific lexicons were used.

Hu and Liu [61] proposed a method for identifying the sentiment of product reviews based on the presence of specific keywords and phrases. They introduced a rule-based approach using POS tagging and dependency parsing to extract aspects from reviews. This approach, known as the lexicon-based approach, has since been widely used in ABSA research. Qiu et al. [62] proposed a method by incorporating the WordNet lexical database into the model to improve the identification of aspects.

In machine learning approaches, different methods have been employed, such as support vector machines (SVM) [63], k-nearest neighbors (k-NN) [64], conditional random field (CRF) [34], hidden Markov models (HMM) [65], naïve Bayes (NB) [66], maximum entropy [67], 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 [68].

There are many surveys related to sentiment analysis. Dang et al. [69] summarized deep learning-based sentiment analysis studies. They analyzed 32 studies that use deep neural networks (DNN), CNNs and recurrent neural networks (RNNs) with Word2Vec embeddings and TF-IDF. Kastrati et al. [70] 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 et al. [71] employed a BiLSTM model with BERT embeddings in Twitter messages. Hung [72] 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. [73] 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 et al. [74] 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. [75] 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. [76] 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. Metalearning 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 eight-shot. Chi et al. [77] used pretrained BERT with fine-tuning and obtained 95.79 percent accuracy. Qizhe et al. [78] 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 2.1 lists the studies related to this dataset.

Table 2.1. Summary of related works on IMDB movie reviews dataset.

Dataset	**Feature Set**	**Model**	**Accuracy**
Haonan et al. (2019) [75]	BERT-Large	Graph Neural Net	96.0
Wang et al. (2021) [76]	RoBERTa-Large	Few-shot learning	96.1
Chi et al. (2019) [77]	BERT-Large	CNN	95.79
Qizhe et al. (2019) [78]	BERT + UDA	BiLSTM	95.8

For the Semeval 2016 Restaurant Reviews dataset, Khalil et al. [79] used Kim's CNN model [80] with Word2Vec embeddings. They used ensemble models with simple voting. Kumar et al. [81] used SVM as the classifier. The features are word polarities retrieved from lexicons, token unigrams and bigrams, and entity-attribute pairs. Brun et al. [82] used a syntactic parser [83] to extract POS tagging, lemma, and surface form of the tokens.

The polarities are found by using an ensemble of singular value decomposition (SVD) [84] and elastic net [85]. Wallaart et al. [86] combined domain knowledge in the form of an ontology and a neural rotatory attention model [87] and used GloVe as word embeddings. Trusca et al. [88] extended the model of Wallaart et al. [86] and used hierarchical attention by adding an extra attention layer to the hybrid approach and replaced GloVe embeddings with ELMo and BERT. Natesh et al. [89] developed a multi-headed self-attention model with the last five layers of the fine-tuned BERT (BERT-IL). Table 2.2 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 2.2. Summary of related works on semeval 2016 restaurant reviews dataset.

Dataset	**Feature Set**	**Model**	**Accuracy**
Khalil et al. (2016) [79]	Word2Vec	CNN	85.49
Kumar et al. (2016) [81]	Lexicon	SVM	86.73
Brun et al. (2016) [82]	Lexicon + CRF	SVD + Elastic Net	88.13
Wallaart et al. (2019) [86]	GloVe	Ontology + LCR-Rot	88.00
Trusca et al. (2020) [88]	ELMo + BERT	Hierarchical Attention	87.00
Natesh et al. (2020) [89]	BERT-IL	Self Attention	88.70

For the Semeval 2017 Twitter dataset, Cliche [90] 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. [91] employed Recurrent-CNN (RCNN) with GloVe and Word2Vec embeddings. RCNN integrates recurrent networks with cohesive convolutional models [92]. Hussam [93] used CNN with structured skip-gram embeddings. Baziotis et al. [94] used bidirectional LSTM (BiLSTM) with GloVe and Word2Vec embeddings. Table 2.3 summarizes the related studies and shows their accuracies.

For the Semeval 2016 Turkish Restaurant Reviews dataset, similar to their experiments on the English Restaurant Reviews dataset, Kumar et al. [81] used word polarities, token n-grams, and entity-attribute pairs as features in an SVM classifier. Ruder et al. [95] used CNN [80] 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 2.4 summarizes the related works on this dataset. Similar to the English counterpart of this dataset, the lexicon-based model yields high performance.

Table 2.3. Summary of related works on Semeval 2017 Twitter dataset.

Dataset	**Feature Set**	**Model**	**Accuracy**
Cliche (2017) [90]	Word2Vec/GloVe/FastText	CNN + LSTM	65.8
Yin et al. (2017) [91]	Word2Vec/GloVe	RCNN	66.4
Hussam (2017) [93]	Word2Vec	CNN	65.2
Baziotis et al. (2017) [94]	Word2Vec GloVe	BiLSTM	65.1

Table 2.4. Summary of related works on Semeval 2016 Turkish restaurant reviews dataset.

Dataset	**Feature Set**	**Model**	**Accuracy**
Kumar et al. (2016) [81]	Lexicon	SVM	84.23
Ruder et al. (2016) [95]	GloVe	CNN	74.21

Finally, for the Beyazperde Movie Reviews dataset, Ucan et al. [96] used a lexicon and SVM to predict the sentiments of the reviews. They obtained 84.6 percent accuracy. Aydın et al. [97] combined Word2Vec with lexicon-based polarity scores to obtain features and used SVM as the classifier. In another work, Aydın et al. [32] 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) [98] 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 2.5.

Table 2.5. Summary of related works on Beyazperde movie reviews dataset.

Dataset	**Feature Set**	**Model**	**Accuracy**
Ucan et al.(2016) [96]	Lexicon-based	SVM	84.6
Aydın et al.(2020) [97]	Lexicon-based + Word2Vec	SVM	90.38
Aydın et al.(2021) [32]	Lexicon + TFIDF	PMI + NN	91.17

There are also several other sentiment analysis studies for the Turkish language. Kaya et al. [99] used maximum entropy, n-gram language model, SVM, and naïve Bayes to classify sentiments of Turkish political columns. Çetin et al. [100] employed active learning for sentiment analysis on Turkish tweets. Türkmenoğlu et al. [101] translated an English sentiment lexicon into Turkish and employed both lexicon-based and machine learning-based methods. Vural et al. [102] customized the SentiStrength sentiment analysis library by translating its lexicon into Turkish and using an unsupervised learning model. Dehkharghani et al. [103] created SentiTurkNet, a Turkish polarity lexicon for sentiment analysis.

Table 2.6. List of embedding methods/features and the learning models used in related studies.

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

To provide a summary of the previous work covered in this section, we show in Table 2.6 the embedding methods/features and the learning models used in these works.

2.2. Aspect Extraction

In recent years, there has been a significant amount of research on aspect-based sentiment analysis, and several different approaches and methods have been proposed. In this section, we review some of the key research papers and developments in this domain.

Early ABSA studies employed rule-based methods for aspect extraction. Various techniques and resources such as pattern matching, syntactic parsing, and domain-specific lexicons were used. Hu and Liu [61] proposed a method for identifying the sentiment of product reviews based on the presence of specific keywords and phrases. They introduced a rule-based approach using POS tagging and dependency parsing to extract aspects from reviews. This approach, known as the lexicon-based approach, has since been widely used in ABSA research. Qiu et al. [62] proposed a method by incorporating the WordNet lexical database into the model to improve the identification of aspects.

Following these works, many researchers have proposed more advanced techniques for ABSA. Some of the works focused on using machine learning algorithms to identify aspects and their corresponding sentiments in the text. As an example in this direction, Jihan et al. [104] proposed an SVM-based approach using the mean embedding feature to extract all related aspect categories. After spell correction, they used lemmatized words and then used mean vectors of Word2Vec embeddings of the lemmas as the feature of their SVM model. In addition to classical machine learning approaches, deep learning models have also begun to be used for aspect extraction tasks. Different models based on recurrent neural networks (RNNs) [105] and convolutional neural networks (CNNs) [106] have been proposed.

Poria et al. [107] introduced a CNN-based model for aspect extraction using semantic and sentiment information. They used Word2Vec embeddings of words and part of speech tags (POStags) as the feature for their CNN model, and also combined their CNN model with a rule-based algorithm (linguistic patterns). Wang et al. [108] proposed an LSTM-based approach that uses attention mechanisms to identify important aspects and their associated sentiments. The input of LSTM model was word embeddings, and the output of their LSTM model fed the attention mechanism. The attention mechanism was used to take into account different parts of the sentence for different aspects. In a recent survey, Chauhan et al. [2] summarized aspect-based sentiment analysis studies that use deep learning models.

Attention mechanisms and transformer architectures have made a significant leap in aspect extraction. As a pioneering transfer model, BERT introduced contextualized embeddings that improved aspect extraction by considering surrounding words and their impact on aspect terms. The transfer learning paradigm was harnessed by fine-tuning pre-trained models like BERT. In a work that followed this approach, the BERT model was fine-tuned for aspect extraction, demonstrating the benefits of transfer learning in capturing domain-specific nuances [109].

Hybrid approaches gained traction in aspect-based studies, combining rule-based, supervised, and deep learning techniques. Chauhan et al. [110] presented a hybrid and domain-based model that integrates rule-based methods and neural networks, achieving improved aspect extraction accuracy. The single word and multi-word aspects are first extracted using a set of linguistic rules. The aspects are pruned based on the frequency and semantic similarity criteria to eliminate the terms that are not relevant to the domain. Then, a BiLSTM network is trained using the extracted aspect terms.

Another vital aspect of ABSA is its multilingual and cross-lingual nature. As businesses operate increasingly globally, there is a growing need for ABSA techniques that can handle multiple languages.

To this end, researchers have proposed various methods, such as cross-lingual transfer learning [111] and multilingual neural networks [112] to improve ABSA performance in different languages. Li et al. [111] developed an improvement transferability model that focused on increasing attention to domain-invariant features while reducing attention to domain-specific features. They tried to determine how closely related different tokens are regarding the type of inference. They proposed a kernelized model with a kernelized attention mechanism that uses a shortest-path graph kernel to constrain the attention scope of variables. They used a metagrammar representation for input features with BERT embedding, and they assembled kernelized and non-kernelized models. Ruder et al. [112] first collected 10,000 most frequent words as the vocabulary for each language. Then they used GloVe embedding vectors for English and random embedding vectors for other languages. They trained a CNN model to extract aspects.

To evaluate aspect extraction methods, benchmark datasets have been widely used. The SemEval datasets [46, 48] provide a standardized basis for performance comparison. Metrics such as precision, recall, F1 score, and accuracy are commonly used for evaluation.

As a summary of the literature review, we observe that aspect extraction is a crucial first step in aspect-based sentiment analysis, affecting the accuracy and granularity of sentiment analysis. The evolution from rule-based methods to deep learning methods, mainly influenced by pre-trained language models like BERT, has transformed aspect extraction. Using hybrid techniques and integrating contextual and linguistic information into the models are promising directions to improve the accuracy of aspect extraction.

3. DATASETS

In Chapters 4 and 5, three datasets in English and two datasets in Turkish were used to analyze the effects of different model architectures and tokenization methods on both languages.

Table 3.1. Number of reviews in the datasets.

	Training			**Test**		
Dataset	**Positive**	**Negative**	**Neutral**	**Positive**	**Negative**	**Neutral**
IMDB Movie Reviews [57]	12500	12500		12500	12500	–
Semeval 2016 Restaurant Reviews [48]	1478	513	57	510	139	27
Semeval 2017 Twitter [113]	19902	7840	22591	2375	3972	5937
Semeval 2016 Turkish Restaurant Reviews [48]	742	469	91	105	39	4
Beyazperde Movie Reviews [96]	13350	13350	–	13350	13350	–

The English datasets are the Stanford IMDB Movie Reviews dataset [57], Semeval 2016 Task 5 Restaurant Reviews dataset [48], and the Semeval 2017 Task 4 Twitter dataset [113]. The Turkish data sets are the Semeval 2016 Task 5 Restaurant Reviews dataset [48] and the Beyazperde Movie Reviews dataset [96]. Table 3.2 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 3.2 in order to be able to compare our results with the results in the literature. Furthermore, 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 datasets, learning models and tokenization methods.

Table 3.2. Number of reviews in the datasets.

Datasets	**Training**	**Test**	**Total**
Semeval 2016 English-Translated Restaurant Reviews [48]	2000	676	2676
Semeval 2016 Turkish Restaurant Reviews [48]	1104	144	1248
Total	3104	820	3924

Table 3.3. Aspects of English and translated(Turkish) reviews.

English ***Translated***	**Reviews**	**Aspects**	**Positions**
English	Judging from previous posts this used to be a good place, but not any longer.	place	11
Translated	*Önceki gönderilerden yola çıkarak bu iyi bir yerdi, ama artık değil.*	yerdi	8
English	I had the duck breast special on my last visit and it was incredible.	duck breast special	4,5,6
Translated	*Son ziyaretimde ördek göğsü özel vardı ve inanılmazdı.*	ördek göğsü özel	3,4,5
English	However, it's the service that leaves a bad taste in my mouth.	service	4
Translated	*Ancak, ağzımda kötü bir tat bırakan hizmettir.*	hizmettir	7

In Chapter 6, two datasets were used; one is in Turkish, and the other is translated from English to Turkish to analyze the effects of different model architectures and embedding on Turkish reviews. These two datasets were created for Semeval 2016 Task 5 [48]. The translated data set is the Semeval 2016 Task 5 Restaurant English Reviews dataset [48], and the Turkish data set is the Semeval 2016 Task 5 Restaurant Turkish Reviews dataset [48]. Table 3.2 shows the number of reviews in the sentiment classes in the datasets. These datasets are skewed datasets made up of three sentiment classes. Some recent studies [114–116] still have used them, as manual labeling of aspects is tough work. We share these datasets on Github[2] .

To translate English reviews, we use Googletrans[3] , a Python library that implements the Google Translate API to access the machine translation service Google Translate[4] .

Several processing operations are performed after the translation. Although many operations are performed, we include only the important ones to guide future studies in translating datasets between different languages:

- Google translate service translates the same word into different words in different reviews. It is an essential issue for embedding vectors and aspect labeling. Therefore, we standardize it, choose one, and replace the others with the chosen one. For example, the word "food" sometimes is translated into "yemek" and sometimes "gıda." This word is crucial for restaurant reviews. Therefore, we replace all the "gıda" words with the "yemek" word.
- Google Translate Service can translate the word differently depending on whether the word is in the sentence or not. However, these datasets give aspect words outside of the reviews. Hence, for some aspects, we change the terms of the aspect by finding the corresponding word in the sentence.

[2] `https://github.com/alierkan/Turkish-ABSA`
[3] https://pypi.org/project/googletrans
[4] https://translate.google.com/

- Since Turkish is an agglutinative language and English is not, the sentence structures differ. Therefore, the aspect word locations in the reviews are changed after translation. Hence, we relabel the aspect words manually concerning the original English aspects. Table 3.3 includes examples of how aspect words are in different places in the English and Turkish reviews.

4. SENTIMENT ANALYSIS USING AVERAGED WEIGHTED WORD VECTOR FEATURES

In this study[5] , we comprehensively analyze some existing methods to produce semantic polarities from the reviews from different domains and propose two new approaches to produce semantic polarities. Our approach uses weighted word vectors to create feature sets. For this purpose, we used the Word2vec [6] and the GloVe [7] models. These models were widely used in sentiment analysis [117,118]. We ensembled and compared our approaches with existing approaches and analyzed the experimental results of IMBD movie reviews, the Semeval 2016 task data set and Yelp restaurant reviews. We are very close to state-of-the-art performance accuracies with our approaches. We provide the source code, prepare datasets for the model, and train word vectors on Github [6] .

4.1. Model

By using machine learning techniques to learn the polarity of a review, we should represent it with some features. As mentioned above, they may be a bag of words, the number of words in a sentence, or other features.

Word vector models (WVMs) offer a powerful way to represent words in a continuous vector space where words with similar meanings end up close. These models have a rich history in Natural Language Processing (NLP), but they all rely on a fundamental idea known as the Distributional Hypothesis. This hypothesis suggests that words sharing similar contexts in text also share similar meanings. Approaches in WVMs can be broadly categorized into count-based and predictive methods.

[5] © 2024 Plos ONE. Reprinted, with permission [36]
[6] `https://github.com/alierkan/Sentiment-Analysis`

Count-based methods analyze how frequently a word appears alongside its neighboring words in a large text corpus and then condense this statistical information into compact vectors for each word. On the other hand, predictive models aim to predict a word based on its neighboring words, using learned compact embedding vectors as model parameters. There are two prominent word vector algorithms in this domain: Word2vec and GloVe.

Word2vec comprises a family of interconnected models designed to generate word embeddings. These models are relatively simple, consisting of two layers of neural networks, and their primary goal is to reconstruct the linguistic contexts in which words appear. The input to Word2vec is typically a sizable collection of text, from which it constructs a vector space, often with dimensions in the hundreds. Each word in the text corpus is assigned a corresponding vector within this vector space. The positioning of these word vectors is such that words sharing similar contextual surroundings in the text data are positioned close to each other within the vector space (Mikolov, 2013).

Our study utilized the Word2vec algorithm, specifically the Continuous Skip-gram Model. This model endeavors to maximize its ability to predict one word based on another word occurring in the same sentence. To achieve this, it treats each word in the text as an input to a log-linear classifier with a continuous projection layer. The model then aims to predict the words likely to appear within a specific range before and after the current word. It is important to note that expanding this range can enhance the quality of the resultant word vectors, but it also introduces increased computational complexity. GloVe is an unsupervised learning algorithm designed to derive vector representations for words. Its training process involves analyzing collective global statistics of word co-occurrence within a corpus, and the resulting representations reveal intriguing linear relationships within the word vector space. GloVe operates by training on the nonzero entries of a global word-word co-occurrence matrix, which captures the frequency with which words appear together in a given corpus. To populate this matrix, only a single pass through the entire corpus is necessary to gather the required statistical information.

By using Word2vec and GloVe algorithms, we produced word vectors separately for the words in the reviews. That is, we have two-word vector sets for every review dataset. We obtained the best results with our methods when the lengths of the Word2vec and GloVe vectors were 300. We eliminated the stop words and did not use word vectors of the stop words. We used these vectors within our weighted averaged review vector (WARV) and the convolutional neural network (CNN) architecture described by Kim [119]. In the following sections, we will explain the methods in detail.

4.1.1. Averaged Review Vector (ARV) and Weighted Averaged Review Vector (WARV)

Since word vectors represent semantic similarity between words, if we can produce a vector that represents semantic similarity for a review or sentence from word vectors, then it will be easy to learn the semantic polarity of the reviews or sentences. For that purpose, if we find a mean vector of a review from word vectors of that review, we have a new vector representing the review, which will be semantically similar to the words in the review. Therefore, we created review vectors from normalized word vectors of words in the review by averaging the word vectors. In this case, all word vectors of the words in a review will be an input of our method, and the output will be a review vector with the same dimension as word vectors. For every review, we found averaged vectors of all reviews. We produced word vectors using the Word2vec and GloVe algorithms. Then we created averaged review vectors (ARV) from Word2vec and GloVe vectors separately and concatenated them. We produced new combined word vectors from Word2vec vectors of size M and GloVe vectors of size M. Obviously, the dimension of the combined word vectors is $2M$. For any review that contains N words, we have averaged N word vectors with size M: $[\mathbf{wv}_1, \mathbf{wv}_2, ..., \mathbf{wv}_N]$. Mathematically, we found normalized vectors of the word vectors by

$$\hat{\mathbf{wv}}_i = \frac{\mathbf{wv}_i}{|\mathbf{wv}_i|} \tag{4.1}$$

where

$$|\mathbf{wv}_i| = \sqrt{wv_{i1} * wv_{i1} + wv_{i2} * wv_{i2} + ...wv_{iM} * wv_{iM}}. \tag{4.2}$$

Then, we found the weighted averaged vectors of normalized word vectors of a review from Word2vec vectors and GloVe vectors separately and concatenated them. For that purpose, we used the following equation, which finds the weighted average of the N word vectors. Note that we will obtain averaged review vectors for $b_i = 1$ for every i with

$$\mathbf{warv}_i = \frac{b_i * \mathbf{wv}_i}{N} \tag{4.3}$$

where b_i represents weights for word vectors. Note that we will obtain averaged review vectors for $b_i = 1$ for every i. We used different weights and obtained the best result with the weight

$$b_i = \frac{P(w_i \geq 0)}{P(w_i \leq 0)} \tag{4.4}$$

where we represented the probability of the positive polarity of the word i as $P(w_i \geq 0)$ and the probability of the negative polarity of the word i as $P(w_i \leq 0)$. To calculate these probabilities, we count number of words in positive sentiment reviews and in negative sentiment reviews of training datasets. For example if a word six times exist in positive reviews, and four times in negative reviews, then $P(w_i \geq 0)$ will be 0.6 and $P(w_i \leq 0)$ will be 0.4. Obviously,

$$P(w_i \geq 0) + P(w_i \leq 0) = 1. \tag{4.5}$$

In this way, we obtain the averaged vectors shown in Table 4.1 for each review using Word2vec and GloVe vectors of the reviews as shown in Table 4.2. Hence, for every review, we have $2M$ features to learn. Using Feed-forward Neural Networks, we learned the review's sentiment with these weighted averaged review vectors (WARV). In the next session, we shared our results for different datasets. We used the Keras framework on Tensorflow to run our feed-forward neural network model, which is shown in Fig 4.1.

We ran our model with different numbers of hidden layers and nodes, different activation, optimizers, and loss functions; however, we obtained the best results with the following parameters: Our model includes three hidden layers. The first two layers have $2M$ nodes.

We used the rectifier (max value of input nodes) "RELU" as an activation function. At the last layer, we used the "sigmoid" function. Our optimizer is "Adadelta," and the loss function is "binary cross-entropy."

Table 4.1. Reviews with ids (M=6).

Review ID	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	...
0	-0.773403	0.080932	0.285580	0.016509	0.379708	...
1	0.420276	0.270612	-0.784823	-0.252692	-0.089924	...
2	-0.312554	-0.435228	-0.589917	0.268841	0.682278	...
3	-0.422671	-0.821369	-0.820449	0.195634	-0.238884	...
4	-0.154693	0.000538	-1.029475	0.405317	-0.271666	...
5	-0.646499	-0.795613	-0.502919	-1.153179	-0.647323	...

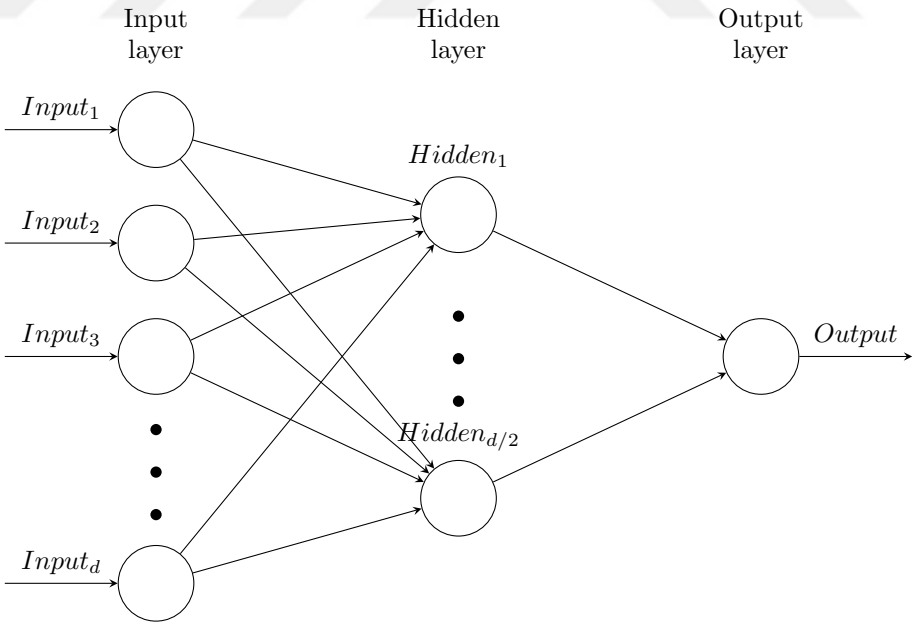


Figure 4.1. Our feed-forward neural network model.

4.1.2. Convolutional Neural Network (CNN)

We used Convolutional Neural Networks to learn the sentiment of the reviews by using Word2vec vectors and GloVe vectors shown in Table 4.2. We used word vectors as features directly. Our learning system is based on the Deep Convolutional Neural Network (CNN) architecture described by Kim [119]. Our architecture is shown in Fig 4.2.

A review matrix is built for each input review, where each row is a vector representation of the word in the review. The review length is fixed to the maximum length of the dataset so that all review matrices have the exact dimensions. (Shorter reviews are padded with row vectors of 0s accordingly.) Each row vector of the review matrix comprises columns corresponding to Word2vec and GloVe vectors concatenated together.

For every word in the review, one embedding vector is used, and all vectors are concatenated for each review. In this way, we obtained a two-dimensional tensor as input of the CNN. Additionally, padding vectors are added for reviews whose tokens are less than the tokens of the largest review. Then, a set of filters is applied to these input tensors of all reviews to produce a feature map.

Next, we employ a max-over-time pooling operation over the feature map, a technique introduced by Collobert et al. (2011) [120]. This operation entails identifying the maximum value within the feature map, denoted as $\hat{c} = \max \mathbf{c}$. The underlying concept is to capture the most significant feature in each feature map by selecting the one with the highest value. This pooling method naturally accommodates varying lengths of reviews, ensuring that one feature is extracted from each filter. The model incorporates multiple filters with different window sizes to derive a range of distinct features. These features collectively constitute the penultimate layer of the model and are subsequently passed to a fully connected softmax layer.

The output of this softmax layer represents the probability distribution over different labels, enabling the model to make classifications based on the learned features.

For every review, we produced review matrices whose rows represent word vectors obtained by concatenation of Word2vec and GloVe as shown in Table 4.2. If one word does not exist in Word2vec/GloVe vectors, then we fill into cells corresponding to Word2vec/GloVe columns with zeros. Also, the number of rows of the review matrices is fixed to some words of maximum-length review (N). For the reviews with fewer words than N, empty rows are filled with zeros. Our Word2vec and GloVe vectors are 300; therefore, we have an input vector whose length is 600 ($= 300 + 300$).

Table 4.2. IMDB reviews with ids.

No	**Word**	**Word2vec 1**	**...**	**Word2vec 300**	**GloVe 1**	**...**	**GloVe 300**
1	storm-lashed	0.536650	...	-0.047624	0.024453	...	0.418517
2	lended	0.182570	...	0.119859	0.203666	...	0.408951
3	maries	0.309272	...	-0.293730	0.374721	...	-0.165671
4	sinfully	0.304134	...	-0.236288	0	...	0
5	weekend	0.436948	...	0.453432	0.038075	...	-0.087220
6	bernie's'	0.269693	...	0.545252	0.094293	...	-0.117270
7	chique	-0.238754	...	0.708557	0.062137	...	0.267371
8	spiritualistic	0.211577	...	0.226885	0.304304	...	0.006760
9	half-awake	0	...	0	-0.093967	...	0.564555
10	eddi	-0.260920	...	0.248055	-0.065446	...	-0.259685
...	...	0	...	0	0	...	0
N	...	0	...	0	0	...	0

We used the Keras/Tensorflow [121] framework to run CNN. In this framework, by default, the filters W are initialized randomly using the glorot uniform method.

Again, we used 600 filters, equal to the input's dimension. At each layer of the network, these filters are applied individually.

During the training process, the values within these filters are optimized using backpropagation, a technique in deep learning, leveraging a specific loss function. In our study, we employed the binary cross-entropy loss for tasks that involve positive and negative sentiment classification (utilizing the IMDB dataset as a reference [57]). We used the categorical cross-entropy loss function for cases where the sentiment analysis task included neutral sentiment as well, as seen in the Semeval 2016 dataset [48].

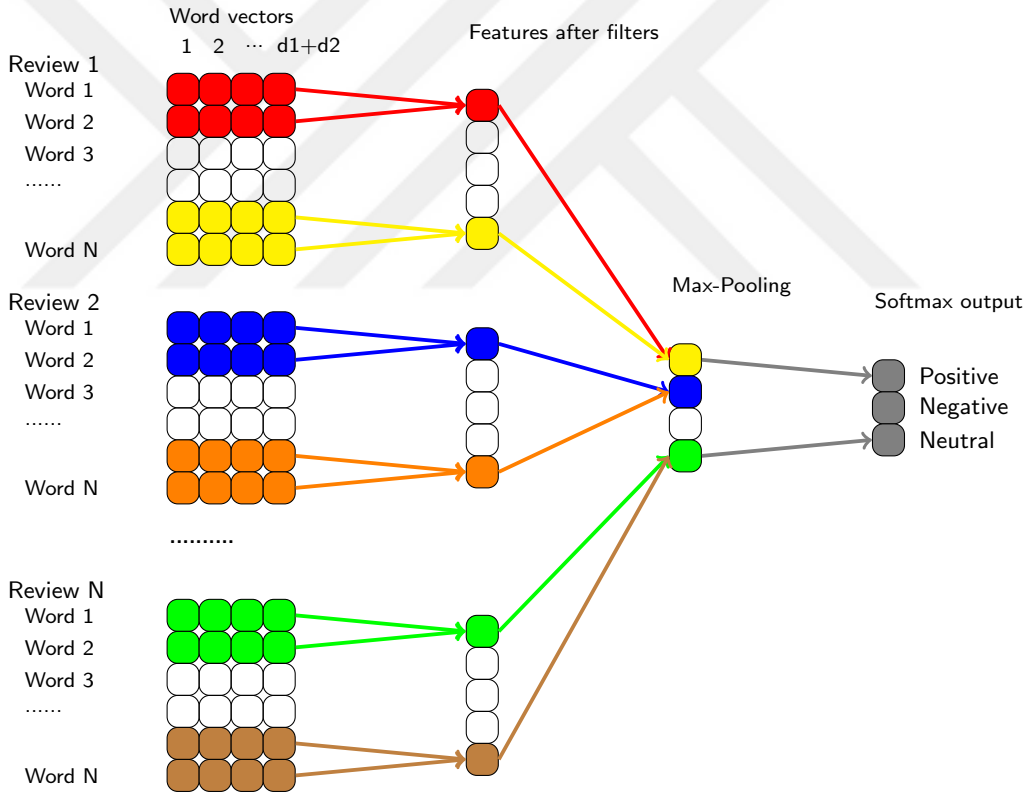


Figure 4.2. CNN sentence classification.

4.1.3. Ensemble

After learning using training data, we combine the results of different learning algorithms using validation data sets. We used two different ensemble approaches:

As a first ensemble approach (Ensemble-1), we used the log probability scores approach in Mesnil's study [59]. As the second one, we used a neural network over validation sets (Ensemble-2) in Fig. 5.1.3. In this case, our learning algorithms produce results by using validation data sets. The results contain probabilities for each class. Then, we used these results as features of the ensemble learning algorithm. For that purpose, we used different learning algorithms to ensemble the methods, and we obtained the best accuracies with logistic regression and neural networks. Our neural network model contains three hidden layers whose number of nodes is equal to the input length. We used Rectified Linear Unit as the activation function of the layers, Sigmoid function to estimate the class or label, AdaDelta [122] as an optimizer, and cross-entropy as a loss function. Then we tested our ensemble methods with test datasets again. Our ensemble method produced better results than every single learning algorithm.

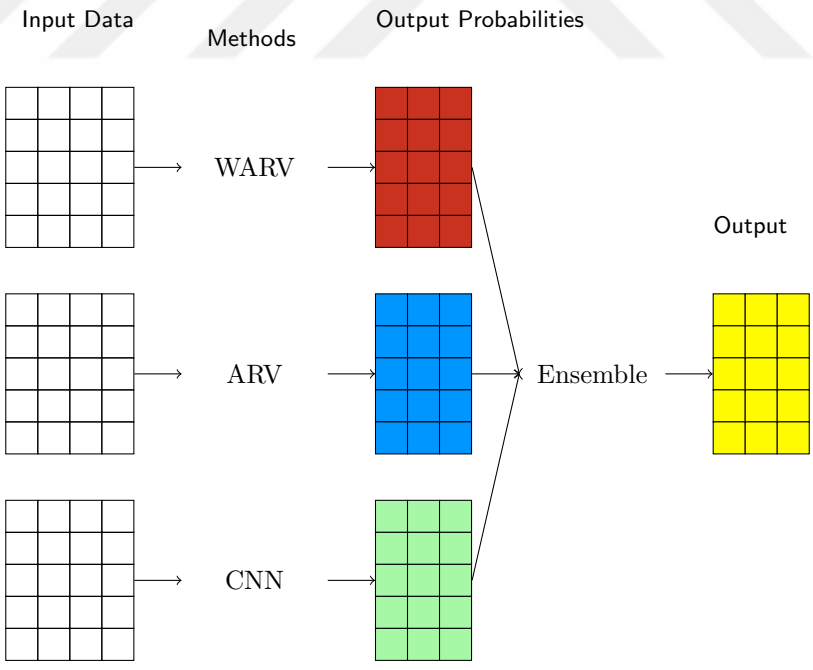


Figure 4.3. Ensemble of learning algorithms.

4.2. Experiments

This work tests our models and other existing models with different datasets. We used the Stanford IMDB Reviews [57], SEMEVAL 2016 Task 5 dataset [48] and the YELP dataset [58]. In this section, we explain experiments and results.

Table 4.3. Accuracies of methods before ensemble for IMDB dataset.

Methods	**Accuracies**	**Explanation**
RNNLM	86.30	Recurrent Neural Network (Mesnil et al. [59])
PV	88.45	Paragraph Vectors (Mesnil et al. [59])
NBSVM	91.87	NB-SVM TriGram (Wang et al. [59] [60])
CNN	95.79	BERT-Large [123]
BiLSTM	95.8	BERT + UDA [124]
Few-shot learning	96.1	RoBERTa-Large [125]
Graph Neural Net	96.0	BERT-Large [126]
ARV_wv	88.05	Average Vectors of Reviews with NN (Word2vec)
ARV_gl	84.54	Average Vectors of Reviews with NN (GloVe)
ARV	88.75 ± 0.38	Average Vectors of Reviews with Neural Network (Word2vec + GloVe)
WARV_wv	93.91	Weighted Average Vectors of Reviews with Neural Network (Word2vec)
WARV_gl	86.24	Weighted Average Vectors of Reviews with Neural Network (GloVe)
WARV	94.95 ± 0.08	Weighted Average Vectors of Reviews with Neural Network (Word2vec + GloVe)
CNN	89.19	CNN with Word Vectors

4.2.1. IMDB Reviews

The Stanford IMDB review dataset contains 100,000 movie reviews in English. 25,000 reviews are labeled as positive, the other 25,000 are labeled as negative, and the remaining 50,000 are unlabeled. Table 4.3 shows accuracy results for different methods. IMDB reviews are labeled numbers that are between 1 and 10. However, this dataset contains only reviews with 1 to 4 as negative and reviews with 7 to 10 as positive. Therefore, there are only two labels: positive and negative. For IMDB reviews, we produced Word2vec and GloVe vectors from IMDB reviews. The length of the vectors is 300, and their window sizes are 5. We run our methods using Word2vec vectors(_wv) and GloVe vectors(_gl) separately and combined(_wvgl). Therefore, the number of features of our Averaged Review Vectors (ARV) and Weighted Averaged Review Vectors (WARV) is 300 for separate runs and 600 for combined runs. Similarly, for our CNN model, the number of features is 300 (separate run) and 600 (combined run).

4.2.1.1. McNemar's Test. McNemar's test [127] is usually used to determine if the success rates of two different algorithms when running on the same input data set are significantly different from each other. It is applied to 2×2 contingency tables.

We used this test to compare the results of the methods to check whether both methods produce significantly different results. For this purpose, we compared the methods two by two, and the results are listed in Table 4.4.

As seen in the table, all methods produce significantly different results from each other at the significance level $p = 0.01$. There is only one exception for the RNNLM and CNN methods. Therefore, we may conclude that especially our WARV method improves the accuracy significantly.

Table 4.4. McNemar's test results of the methods.

Method 1	**Method 2**	**Test Statistic**	**p-value**
RNNLM	PARAGRAPH	43.513	0.000
RNNLM	NBSVM	9.948	0.002
RNNLM	ARV	258.930	0.000
RNNLM	CNN	1.099	0.295
PARAGRAPH	NBSVM	19.931	0.000
PARAGRAPH	ARV	156.272	0.000
PARAGRAPH	CNN	86.196	0.000
NBSVM	ARV	265.454	0.000
NBSVM	CNN	27.795	0.000
ARV	CNN	429.937	0.000
WARV	RNNLM	89.331	0.000
WARV	PARAGRAPH	8.853	0.003
WARV	NBSVM	66.513	0.000
WARV	ARV	77.704	0.000
WARV	CNN	155.252	0.000

4.2.1.2. Ensemble. As seen in Table 4.3, with % 94.95 accuracy, our WARV method has better accuracy values than the N-Gram, RNNLM and Mesnil's paragraph vector. Also, our WARV method is computationally more efficient than Mesnil's paragraph vectors (PV) (Mesnil et al. [59]). When our weighted averaged review vectors (WARV) are ensembled with NBSVM, they reach the best accuracy value % 95.56 is close to the state-of-the-art result % 96.10 [125] with RoBERTa-Large embedding as shown in Table 4.5 and 4.3 respectively. When we ensemble all six methods, accuracy becomes % 95.48. These results are nearly close to other results that were obtained by using Bidirectional Transformers BERT embeddings [128] in some studies ([123] [124] [125] [126]) that reached % 96.10 [125].

Note that we did not share the ensemble results of the combinations whose accuracies are below % 94 except for ensembles in Mesnil's study [59].

4.2.2. Semeval 2016 Task 5 Dataset

In English, two domain-specific datasets for consumer electronics (laptops) and restaurants, consisting of more than 1000 review texts (approximately 6K sentences) with fine-grained human annotations (opinion target expressions, aspect categories and polarities) will be provided for training/development. In particular, the SE-ABSA15 train and test datasets for restaurants and laptops (with some corrections) will be made available as training data. They consist of 800 review texts (4500 sentences) annotated with approximately 15000 unique label assignments (Entity, Aspect, Polarity).

The laptop dataset comprises 450 review texts (2500 sentences) annotated with 2923 Entity#Aspect, polarity tuples. The restaurant dataset comprises 350 review texts (2000 sentences) annotated with 2499 Entity#Aspect, polarity tuples. All datasets will be enriched with text-level annotations. In addition, data sets exist for other languages rather than English. These languages are Arabic, Chinese, Dutch, French, Russian, Spanish, and Turkish [48].

We used our two methods in the English restaurant dataset. We produced Word2vec and GloVe vectors from the Yelp dataset. For this purpose, we first get only restaurant-related reviews from the Yelp dataset and then create word vectors. After that, we applied our two methods (ARV and CNN) to the Semeval 2016 Task 5 dataset to learn polarity. Mainly, our ARV method produced high accuracy results (87.7%). It is very close to the best result of Semeval 2016 Task 5 as shown in Table 4.6 and the study [129] (87.8%) with BERT embedding [128]. We ensembled our two results using a neural network. For this purpose, we used the probability output of ARV and CNN. Since we have three classes (positive, neutral, and negative), each method has three probability values. Therefore, we used six probability values of two methods as our learning features.

Since the Semeval 2016 dataset is tiny, we have no opportunity to use some parts of the dataset for validation.

Table 4.5. Accuracies of methods after ensemble for IMDB dataset.

Methods	**Ensemble-1**	**Ensemble-2**
RNNLM, PARAGRAPH	90.14 (Mesnil et al. [59])	86.98 ∓ 1.18
RNNLM, NBSVM	92.08 (Mesnil et al. [59])	91.03 ∓ 0.39
PARAGRAPH, NBSVM	92.13 (Mesnil et al. [59])	91.80 ∓ 0.43
WARV, RNNLM	95.22 ∓ 0.06	94.76 ∓ 0.21
WARV, PARAGRAPH	94.62 ∓ 0.04	94.70 ∓ 0.09
WARV, NBSVM	95.56 ∓ 0.03	94.92 ∓ 0.54
WARV, ARV	94.35 ∓ 0.09	94.70 ∓ 0.23
WARV, CNN	94.44 ∓ 0.07	93.82 ∓ 1.18
RNNLM, PARAGRAPH, NBSVM	92.37 (Mesnil et al. [59])	91.92 ∓ 0.28
WARV, PARAGRAPH, NBSVM	94.59 ∓ 0.02	95.42 ∓ 0.13
WARV, PARAGRAPH, ARV	93.25 ∓ 0.07	94.93 ∓ 0.17
WARV, PARAGRAPH, CNN	93.42 ∓ 0.04	94.99 ∓ 0.22
WARV, NBSVM, ARV	94.45 ∓ 0.06	94.82 ∓ 1.33
WARV, NBSVM, CNN	94.10 ∓ 0.07	95.02 ∓ 0.90
WARV, ARV, CNN	93.37 ∓ 0.07	95.07 ∓ 0.11
RNNLM, PARAGRAPH, WARV	94.59 ∓ 0.04	94.85 ∓ 0.15
RNNLM, WARV, ARV	94.42 ∓ 0.07	94.82 ∓ 0.21
RNNLM, WARV, CNN	94.22 ∓ 0.06	94.94 ∓ 0.15
RNNLM, NBSVM, WARV	95.18 ∓ 0.02	95.34 ∓ 0.26
RNNLM, PARAGRAPH, NBSVM, WARV	94.47 ∓ 0.03	95.35 ∓ 0.16
RNNLM, PARAGRAPH, ARV, WARV	93.50 ∓ 0.07	94.78 ∓ 0.28
PARAGRAPH, NBSVM, ARV, WARV	93.85 ∓ 0.04	95.31 ∓ 0.18
PARAGRAPH, NBSVM, CNN, WARV	93.89 ∓ 0.04	94.68 ∓ 1.27
PARAGRAPH, WARV, ARV, CNN	93.14 ∓ 0.05	95.11 ∓ 0.06
RNNLM, NBSVM, ARV, WARV	94.45 ∓ 0.06	95.42 ∓ 0.12
RNNLM, NBSVM, CNN, WARV	94.26 ∓ 0.04	95.24 ∓ 0.08
NBSVM, WARV, ARV, CNN	93.92 ∓ 0.05	95.38 ∓ 0.19
RNNLM, WARV, ARV, CNN	93.70 ∓ 0.07	95.06 ∓ 0.09
PARAGRAPH, WARV, ARV, CNN	93.14 ∓ 0.05	94.65 ∓ 0.71
WARV, PARAGRAPH, NBSVM, ARV, CNN	93.57 ∓ 0.05	95.30 ∓ 0.25
RNNLM, WARV, NBSVM, ARV, CNN	94.06 ∓ 0.04	95.38 ∓ 0.10
RNNLM, PARAGRAPH, WARV, ARV, CNN	93.35 ∓ 0.05	95.09 ∓ 0.06
RNNLM, PARAGRAPH, NBSVM, WARV, CNN	94.04 ∓ 0.03	95.37 ∓ 0.15
RNNLM, PARAGRAPH, NBSVM, ARV, WARV	93.91 ∓ 0.04	95.29 ∓ 0.15
RNNLM, PARAGRAPH, NBSVM, ARV, CNN, WARV	93.68 ∓ 0.05	95.48 ∓ 0.04

Table 4.6. Accuracies of methods (Ensemble-2) for Semeval restaurant dataset.

Methods	**Accuracies**	**Explanation**
Self Attention	88.70	BERT-IL [129]
ARV, WARV and CNN	88.669 ∓ 3.60	Ensemble of ARV, WARV and CNN by Neural Network
WARV and CNN	88.630 ∓ 3.81	Ensemble of WARV and CNN by Neural Network
ARV and CNN	88.475 ∓ 1.79	Ensemble of ARV and CNN by Neural Network
XRCE	88.126	Semeval 2016 Task 5 Slot 1 Best Accuracy [48]
Hierarchical Attention	87.00	ELMo + BERT [130]
IIT-T	86.729	Semeval 2016 Task 5 Slot 1 Second Best Accuracy [48]
NileT	85.448	Semeval 2016 Task 5 Slot 1 Third Best Accuracy
IHS-R	83.935	Semeval 2016 Task 5 Slot 1 Fourth Best Accuracy [48]
ECNU	83.236	Semeval 2016 Task 5 Slot 1 Fifth Best Accuracy [48]
WARV	83.95 ∓ 0.38	Weighted Averaged Review Vectors with Neural Network
CNN	83.120 ∓ 0.44	Convolutional Neural Network with Word Vectors
INSIG	82.072	Semeval 2016 Task 5 Slot 1 Sixth Best Accuracy [48]
ARV	82.08 ∓ 0.68	Averaged Review Vectors with Neural Network
basel	76.484	Base of Semeval 2016 Task 5 Slot 1 [48]

Instead of probability values in the validation data, we used probabilities of the training dataset. Our ensemble feed-forward neural network used probabilities produced over the training dataset of Semeval 2016 Task5 by our ARV and CNN methods. Then we used the test dataset of Semeval 2016 Task 5 to evaluate our ensemble model.

As shown in Table 4.6, our ensemble produced results that are very close to the state-of-the-art value (88.242%) for the Semeval 2016 Task 5 dataset. We used the Keras framework to run our ensemble neural network model.

Our ensemble learning neural network includes three hidden layers with 600 nodes, equal to the number of features. As an activation function, we used "scaled exponential linear units" (SELUs), which induce self-normalizing properties [131]. At the last layer, we used the "softmax" function. Our optimizer is "Adadelta," and the loss function is "binary cross-entropy."

Note that we did not share the results of the RNNLM, NBSVM and PV methods, which were compared with our methods in the previous section for this dataset since they suffered from a small dataset. They could not learn accurately, and their results could have been better.

Pretrained transformer-based models have recently been used in sentiment analysis [132]. We also used average pretrained transformer-based models BERT vectors [128] to compare our results with transformer-based models. For that purpose, we got vector tokens in the last layer of the BERT model; then, we calculated the average for each review, which is the same method as our other models. Then, we used these vectors instead of our Word2vec and GloVe vectors to learn the sentiment of the reviews. For the Semeval 2016 dataset, we obtained 86.39% accuracy which is less than our ensemble results. Furthermore, for the IMDB movie review dataset, we obtained 90.43% accuracy, which is less than our ensemble results.

4.2.3. Yelp Dataset

We tested a weighted average review vector with randomly chosen restaurant-related reviews. When we used 100.000 reviews (50.000 positive, 50.000 negative) for training and 100.000 reviews (50.000 positive, 50.000 negative) for testing, our accuracy was % 73.811. When we used 500.000 reviews (250.000 positive, 250.000 negative) for training and 500.000 reviews (250.000 positive, 250.000 negative) for testing, our accuracy was % 77.496. When we randomly selected 100.000 restaurant reviews (50.000 positive, 50.000 negative) and divided the data into 90% for training and 10% for testing, we applied a ten-fold cross-validation. As seen in Table 4.7, ensembling does not increase the accuracies for a randomly chosen Yelp dataset. This means that all methods are able to learn the same reviews correctly and make mistakes for the same reviews.

Table 4.7. Accuracies for Yelp restaurant dataset with 100.000 reviews.

Methods	**Accuracies**	**Explanation**
ARV, WARV and CNN	75.51 ∓ 0.53	Ensemble of ARV, WARV and CNN by Neural Network
WARV and CNN	74.55 ∓ 0.42	Ensemble of WARV and CNN by Neural Network
ARV and CNN	75.12 ∓ 0.37	Ensemble of ARV and CNN by Neural Network
ARV and WARV	75.80 ∓ 0.67	Ensemble of ARV and CNN by Neural Network
CNN	73.22 ∓ 0.90	Convolutional Neural Network with Word Vectors
WARV	75.76 ∓ 0.19	Weighted Averaged Review Vectors with Neural Network
ARV	75.89 ∓ 0.39	Averaged Review Vectors with Neural Network

We produced word2vec and glove vectors with dimension 200 for each using a window size of five. We run our feedforward models using word2vec vectors and glove vectors together, and we got the best results with dimension 200. Thus, the number of features in the models is 200 for the separate runs and 400 for the combined runs. Similarly, in the CNN model, the column dimensions of the input matrices are 200 (separate run) and 400 (combined run).

Again, we used word vectors that were produced from the Yelp dataset. This dataset [58] is a subset of reviews, and user data especially for use in academic purposes. Available in both JSON and SQL files. Each file comprises a single object type, one JSON object per line. According to the business IDs, we divided the review file into two separate files; one of them consists of restaurant reviews, and the other one consists of non-restaurant reviews. For restaurants, we used approximately two million reviews to produce word vectors. For this purpose, we used two files from the Yelp dataset:

- business.json: Business data with location data, attributes, and categories.
- review.json: Review text data.

4.3. Conclusion

In this work, we used a combination of Word2vec and GloVe vectors. These vectors were used directly for the CNN model to create review matrices. For our averaged vector model, for a review, we found averaged vectors from word vectors of the word in the review. For our weighted averaged vector model, we first multiplied the word vectors by predefined values, such as the ratio of numbers of positive and negative words. Then, we found averaged vectors. Then, we used these vectors as input to the learning algorithms. We compared our results with different methods and with different datasets. Moreover, we are very close to the state-of-the-art results for the IMDB dataset and the Semeval-2016 Task-5 dataset although we did not use any pre-trained model such as BERT [128] or the Large Language Model LLAMA [133].

Our weighted averaged vector model obtained high accuracy values for the IMDB dataset and is computationally very efficient. Building word vectors only once is enough for our method, although the paragraph vector method needs to build word vectors for all new input data. Furthermore, our ensemble results are very high, and our ensemble method produced better results than every single learning algorithm. For our method, the next step is a lookup operation to find the word vector of the words and calculate the weighted average of the word vectors. We will continue to evaluate our models and their ensembles with different datasets in different domains. Furthermore, our models do not include language-specific features. Therefore, we will test our model in different languages rather than English. In addition, we will find different weights that may produce better results.

5. DEEP LEARNING MODEL COMBINATIONS AND TOKENIZATION APPROACHES IN SENTIMENT CLASSIFICATION

In this study[7] , we use 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 [10] (XLM-RoBERTa [11] 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) [80]. 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.

[7] © 2023 IEEE. Reprinted, with permission [37]

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) [26], WordPiece [29], unigram language model (ULM) [30], 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 [134] [135]. 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 [136] [137]. The tokenization methods BPE, WordPiece, and ULM are referred to as subword methods and are used mainly in deep learning-based studies [26] [138]. 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 [57], Semeval 2016 Task 5 Restaurant Reviews [48], and Semeval 2017 Task 4 Twitter [113] datasets for English, and Semeval 2016 Task 5 Turkish Restaurant Reviews [48] and Beyazperde Movie Reviews [96] 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[8] .

The main contributions of the study 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.

[8] `https://github.com/alierkan/TokenizedSentimentClassification`

- We conduct a stability analysis for the learning models and the tokenization methods.

5.1. Models

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 following subsections.

As stated in Introduction Chapter 1, 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 6.2. 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.

5.1.1. CNN With Word2Vec And GloVe Embeddings

We generate Word2Vec and GloVe embeddings from our datasets. We also use the YELP dataset [58] to generate embeddings for the Semeval 2016 restaurant dataset. Figure 5.1 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 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) [**?**]. 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.

5.1.2. 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 5.2.

5.1.3. CNN With BERT/RoBERTa/XLM-RoBERTa Embeddings

This model is the same as the model in Section 5.1.1, with the difference that BERT and RoBERTa (XLM-RoBERTa) embeddings are used instead of Word2Vec and GloVe embeddings. Figure 5.3 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.

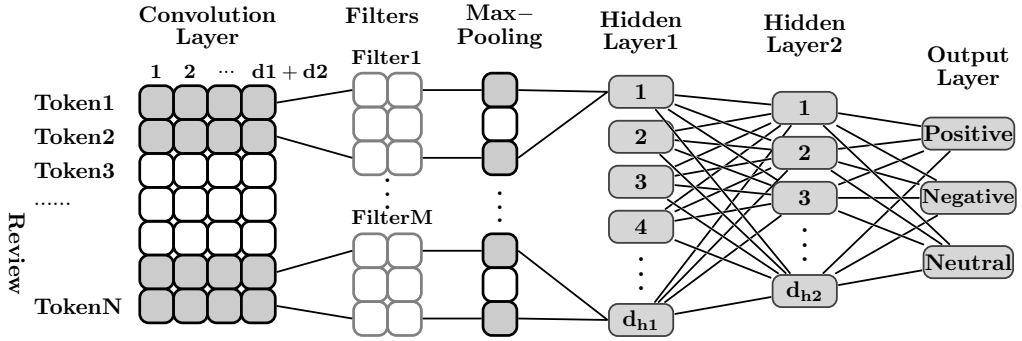


Figure 5.1. CNN model with Word2Vec and GloVe embeddings.

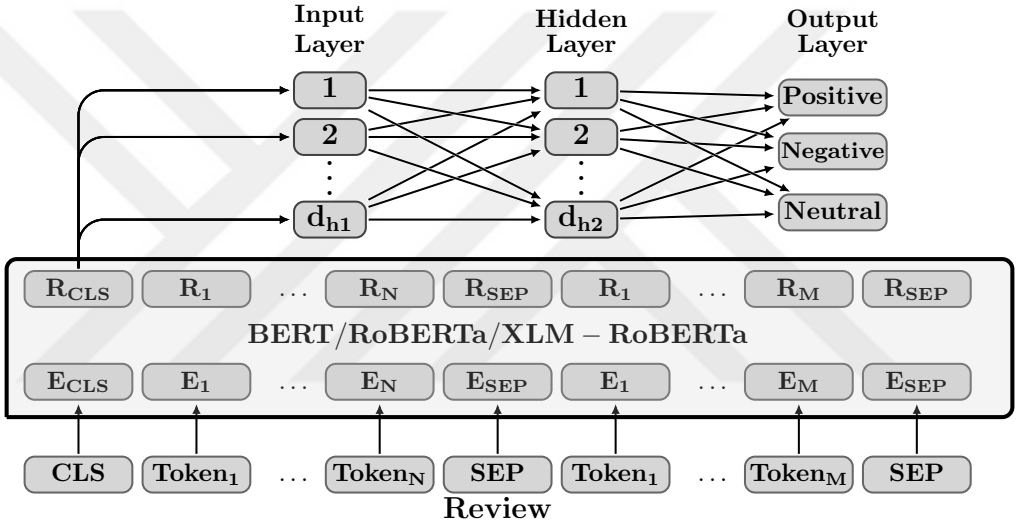


Figure 5.2. FFNN model with Bert/RoBERTa/XLM-RoBERTa embeddings.

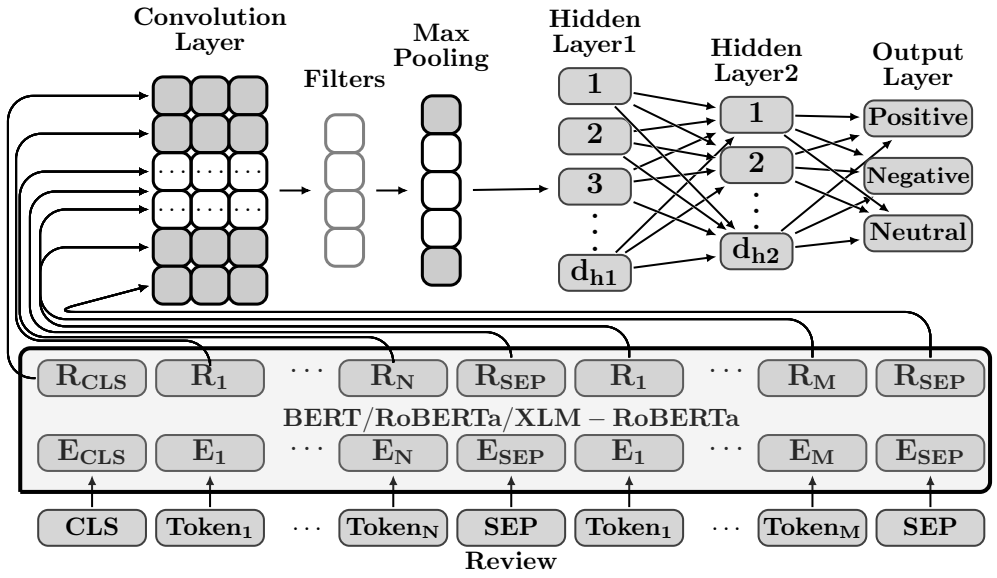


Figure 5.3. CNN model with Bert/RoBERTa/XLM-RoBERTa embeddings.

5.2. Experiments

In this section, we show the performance of the models explained in Section 6.1 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 5.1. 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 5.1 and 5.2 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 5.1.1). The results on the right of the table correspond to the FFNN (Section 5.1.2) and CNN (Section 5.1.3) 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 5.1 are also visualized in Figure 5.4 to facilitate comparisons with respect to tokenization methods and models.

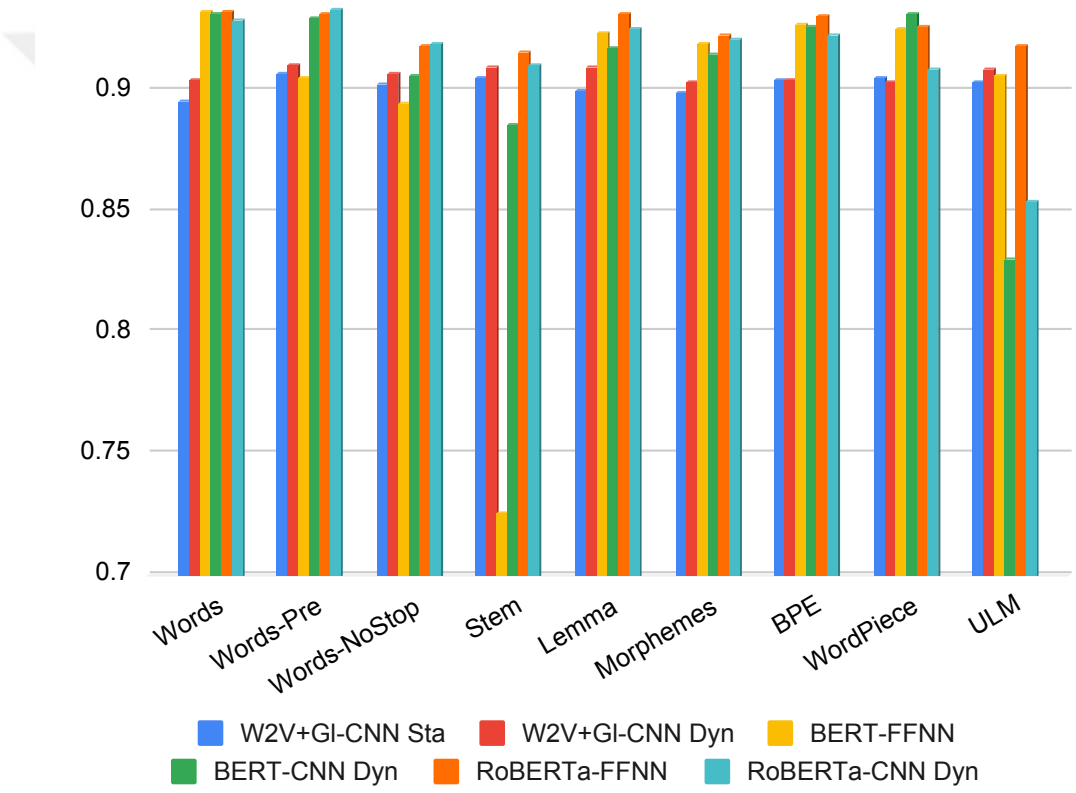


Figure 5.4. Accuracy results for Stanford IMDB movie reviews dataset.

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 5.2. 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

Table 5.3 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 5.4 and 5.5 and Figure 5.5 (corresponding to Table 5.4) show the macro F1-value scores of the models.

Table 5.3. 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-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 5.4. 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

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 5.6, 5.7, and 5.8 and in Figure 5.6 (corresponding to Table 5.7). 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 5.5. 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

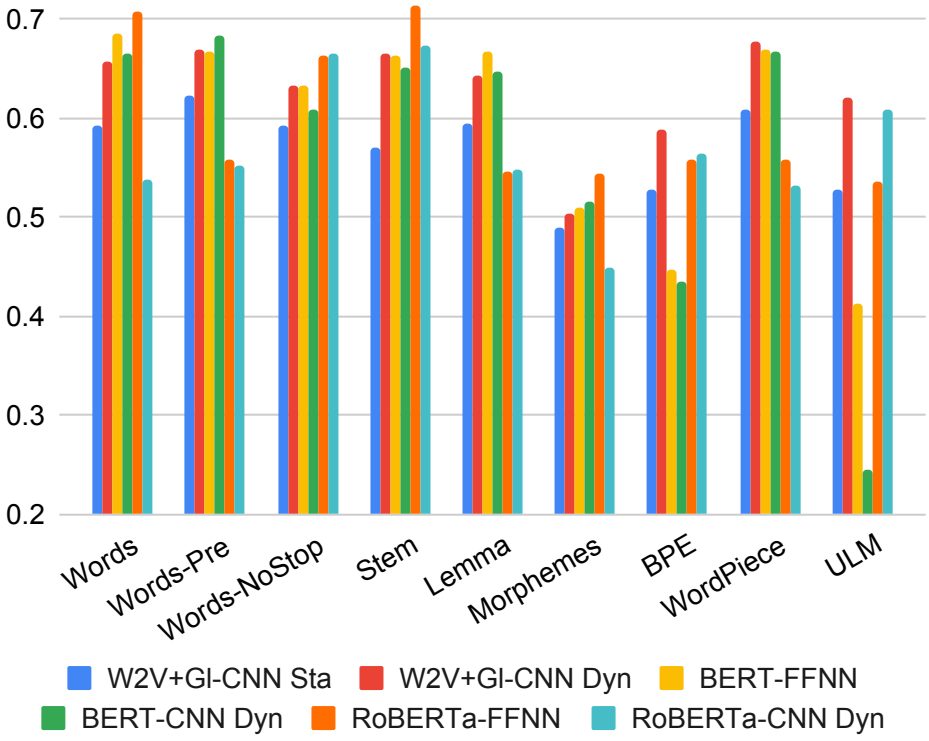


Figure 5.5. Macro F1-values for Semeval 2016 restaurant reviews dataset.

Table 5.6. 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

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

In addition to the three English datasets, we give the results for the two Turkish datasets. Tables 5.9, 5.10, and 5.11 and Figure 5.7 (corresponding to Table 5.10) show the results for the Semeval 2016 Turkish Restaurant Reviews dataset.

Table 5.7. 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

Tables 5.12 and 5.13 and Figure 5.8 (corresponding to Table 5.12) 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.

Table 5.8. 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

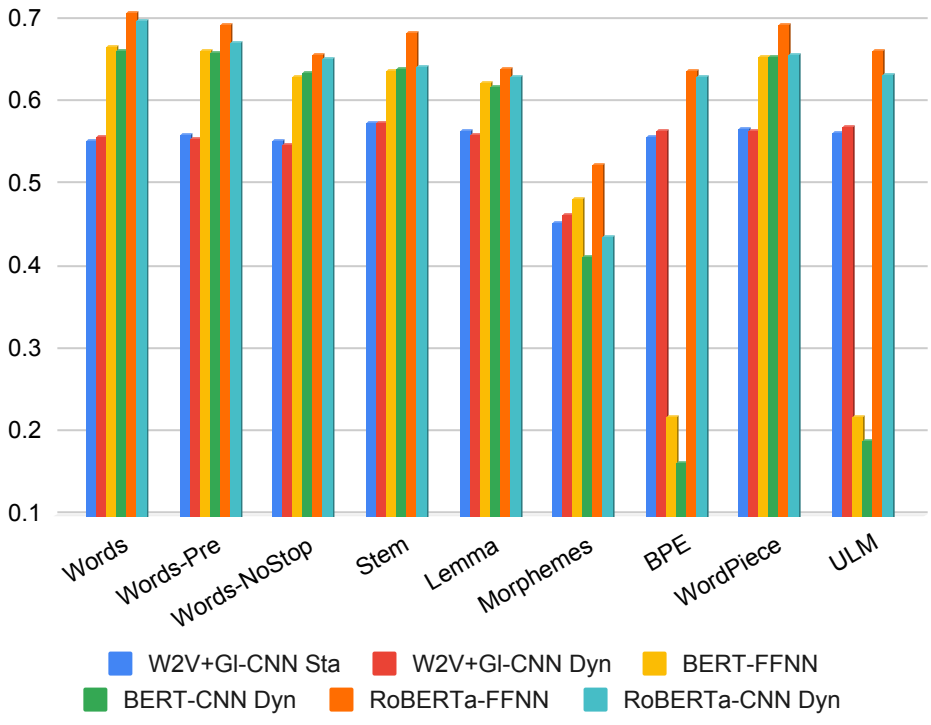


Figure 5.6. Macro F1-values for Semeval 2017 Twitter dataset.

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 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.

Table 5.9. Accuracy results for Semeval 2016 Turkish restaurant reviews dataset.

Embeddings:	**Word2Vec+GloVe**		**BERT-base**		**XLM-RoBERTa**	
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 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.

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.

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.

Table 5.10. Macro F1-values for Semeval 2016 Turkish restaurant reviews dataset.

Embeddings:	**Word2Vec+GloVe**		**BERT-base**		**XLM-RoBERTa**	
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

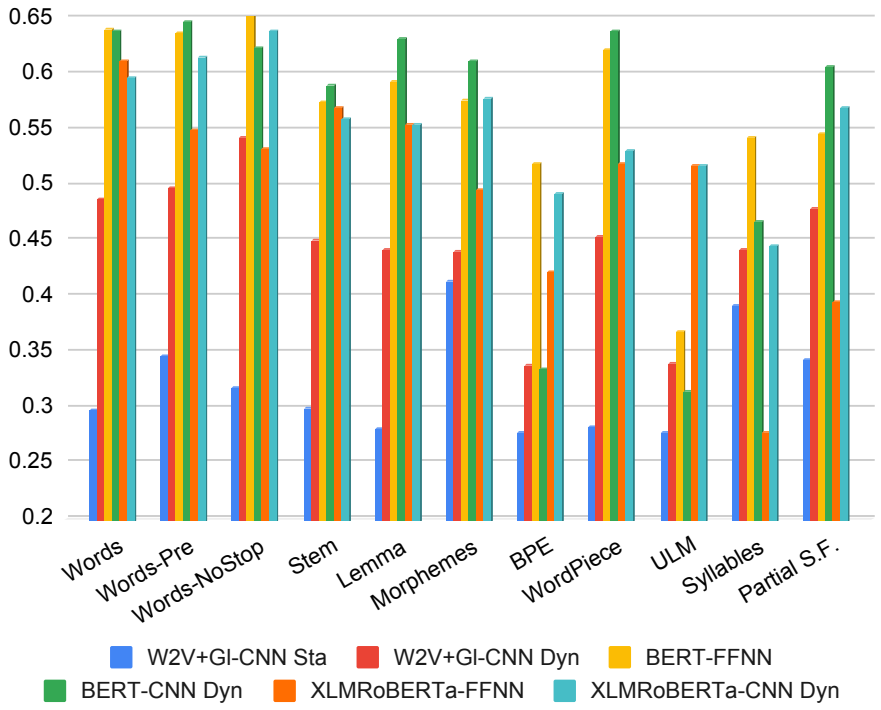


Figure 5.7. Macro F1-values for Semeval 2016 Turkish restaurant reviews dataset.

Table 5.11. K-fold macro F-1 values for Semeval 2016 Turkish restaurant reviews dataset.

Embeddings:	**Word2Vec+GloVe**		**BERT-base**		**XLM-RoBERTa**	
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

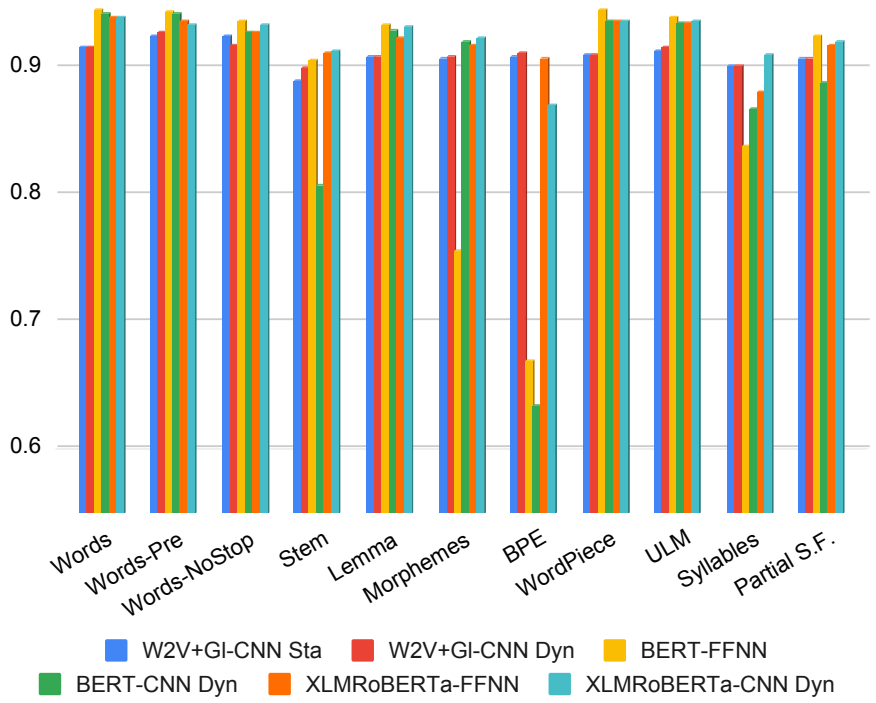


Figure 5.8. Accuracy results for BeyazPerde movie reviews dataset.

Table 5.12. Accuracy results for BeyazPerde movie reviews dataset.

Embeddings:	**Word2Vec+GloVe**		**BERT-base**		**XLM-RoBERTa**	
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

Table 5.13. K-fold accuracy results for BeyazPerde movie reviews dataset.

Embeddings:	**Word2Vec+GloVe**		**BERT-base**		**XLM-RoBERTa**	
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

These tokenization approaches outperform the popular subword methods BPE, 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.

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.[9] The analysis shows how much each model is affected by a change in the tokenization method. Table 5.14 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 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 5.15 shows the standard deviation averaged over the models and the English/Turkish datasets for each method.

[9] 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 5.14. 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**	
		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

Table 5.15. 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

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 5.16 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.

Table 5.16. 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

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.

5.3. 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 5.1-5.13. 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 [139].

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 5.17 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 5.1-5.13. 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 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 6.2, 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 5.17. 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 5.18 shows the accuracy results and the increase over the best outcome for each dataset given in Tables 5.1-5.13 with below token-embedding combinations:

- 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).

Table 5.18. 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.

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)

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 5.18 with those in Table 5.17 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 5.17, but a general ensemble model in Table 5.18.

5.4. 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.

5.5. 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.

6. ASPECT EXTRACTION FROM TURKISH REVIEWS BY USING BERT, BILSTM, CRF, AND TREE POSITIONAL ENCODING

This research is motivated by the recognition that existing ABSA models, while successful to varying degrees, often face challenges in handling the complexity of real-world textual data. By combining state-of-the-art techniques, our approach seeks to push the boundaries of aspect extraction precision and contribute to a more nuanced understanding of sentiments in textual information.

This research focuses specifically on ABSA's aspect extraction component. By shedding light on the importance of accurate aspect extraction, we aim to contribute to the broader fields of sentiment analysis and opinion mining. Improved aspect extraction techniques advance our understanding of user opinions and have practical implications for industries, decision-making processes, and the development of context-aware applications. We also develop a model to learn sentiment based on hidden layer vectors of the aspect extraction model.

6.1. Model

In this chapter, we explain the aspect extraction models used in this work. We employ different model combinations with different types of embeddings. The general architecture is shown in Figure 6.1. The models take as input information about the tokens in the review in the form of three types of embedding, such as word embeddings, POS tag embeddings, and tree positional encodings. The details of the models and the embedding schemes are explained in the following subsections.

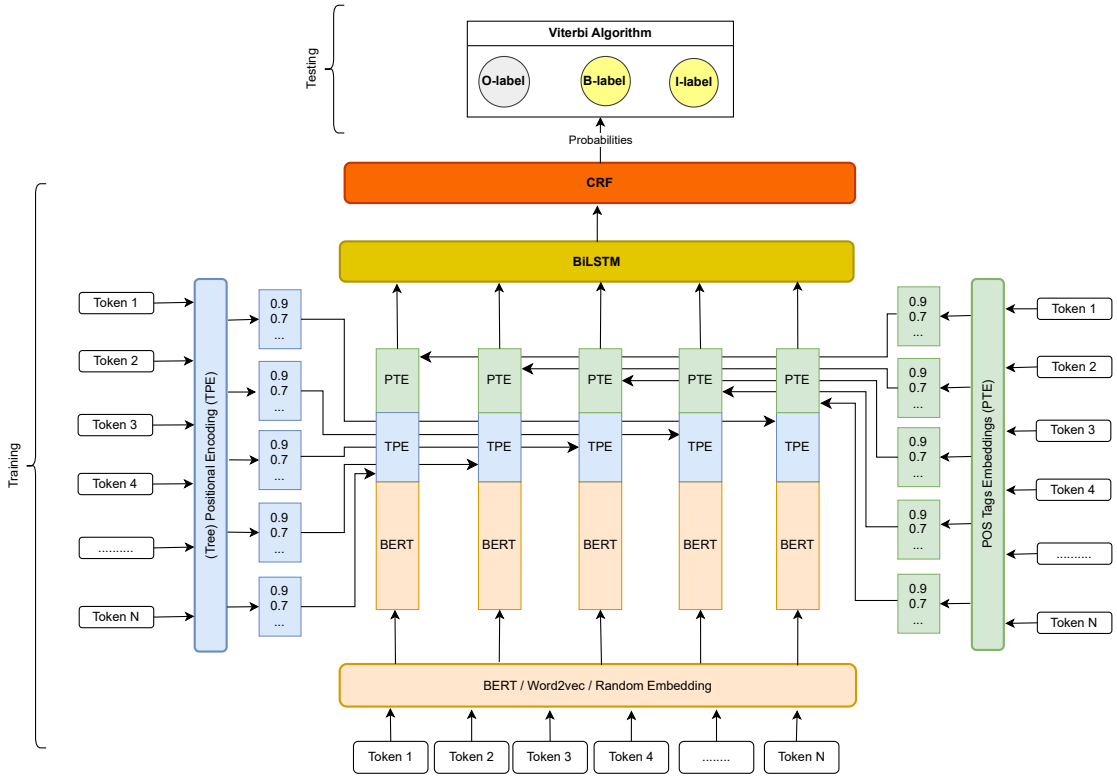


Figure 6.1. (BERT-)BiLSTM-CRF model

6.1.1. Word Embeddings

We build two different models depending on how the word embeddings are fed into the model. In the first model, we use randomly initialized embeddings or domain-specific Word2Vec embeddings as word embeddings for the words in the review. We train domain-specific Word2Vec embeddings using the two Turkish reviews datasets explained in Chapter 3. We build a word embedding repository from these pretrained Word2Vec embeddings which are then used in the aspect extraction models. In both cases (random embeddings and Word2Vec embeddings), the word vectors are initialized with these embeddings and then updated during the training phase of the aspect extraction models. In this way, the word vectors are dynamically adjusted with respect to the aspect extraction task. The token embeddings are initialized with 300-dimensional Word2Vec and random vectors.

In the second model, we use a Turkish BERT model ("bert-base-turkish-uncased"[10]) to produce the word embeddings. Instead of using random or pretrained embeddings as input to the model, we feed the words in the review as input to the BERT model and use the token outputs as word embeddings. As in the first model, the BERT embeddings are updated during the training phase of the aspect extraction models. The BERT base models have 756-dimensional vectors.

6.1.2. POS Tag Embeddings

Part-of-speech tags of the words in reviews are obtained using the BOUN dependency parser[11] [140]. The parser had been trained on three Turkish dependency treebanks in the Universal Dependencies (UD) framework[12] . It uses the POS tagset used in these treebanks, which is formed of 17 POS tags. As in the case of word embeddings, we use randomly initialized or domain-specific Word2Vec embeddings for the POS tag vectors of the words. Domain-specific POS tag embeddings are trained in the same manner as word embeddings. We represent the reviews in the datasets in terms of POS tags of the words, combine the reviews in the two datasets, and then train the Word2Vec embeddings using the Word2Vec algorithm. Random or Word2Vec embeddings are used to initialize the POS tag vectors in the models. The POS tag vectors are then dynamically updated during the training of the aspect extraction models. The POS tags embeddings are initialized with 100-dimensional Word2Vec and random vectors.

6.1.3. Position Encoding

In the original transformer model proposed by Vaswani et al. [141], positional encodings are used in order to capture the position information of words in a sequence. For each word in the sequence, the word embedding vector and the positional embedding vector are added, which is then used as input to the attention layers.

[10]https://huggingface.co/dbmdz/bert-base-turkish-uncased
[11]`https://github.com/BOUN-TABILab-TULAP/Boun-Parser`
[12]https://universaldependencies.org/

The dimension of the positional encoding is set equal to the dimension of the word vector. A static approach is used in forming the positional encoding that gives a unique encoding for each position in the sequence. The following equations that are used in forming the positional encoding of position pos (PE_{pos}) in a sequence are

$$p_{pos,2i} = \sin\left(\frac{pos}{M^{\frac{2i}{d}}}\right) \tag{6.1}$$

and

$$p_{pos,2i+1} = \cos\left(\frac{pos}{M^{\frac{2i}{d}}}\right) \tag{6.2}$$

where d is the dimension of the position vector, $0 \leq i < \frac{d}{2}$ is used to index the elements of the vector, and M is a user-defined constant which was set to 10,000 in the original architecture. As can be seen, the even-numbered elements of the vector are computed using the sine function and the odd-numbered elements using the cosine function.

During our preliminary work in aspect extraction, we observed that the aspect words in Turkish reviews are usually the head words of the phrases in the sentences. Based on this observation and knowing that the head words in the dependency parse of a sentence appear close to the root of the parse tree, we decided to use this information in the aspect extraction models and integrated it into the position information. In this direction, instead of using the word positions in the sentence as positional encodings, we use the level numbers of the words in the dependency parse tree as positional encodings. As an example, the dependency parse tree of the sentence "*Son olarak dün gittiğim her gittiğimde ayrı keyif aldığım güzel mekan . (Finally, I went yesterday, it is a beautiful place that I enjoy every time I go.)*" is shown in Figure 6.2 accompanied with the level indices. The root word is indexed with the length of the path between it and the most distant word (max-index) and the index of a word is set as (max-index - distance to the root). To index the words, we use a recursive algorithm starting with the root word in the dependency parser tree. In the example sentence, we see that the aspect term "mekan" ("place") is associated with level index 6 which differentiates it from non-aspect terms in the sentence that have lower index values.

To integrate the level information into the model, we modified the positional encodings as

$$p_{pos,2i} = \sin\left(\frac{level_index(pos)}{M^{\frac{2i}{d}}}\right) \tag{6.3}$$

and

$$p_{pos,2i+1} = \cos\left(\frac{level_index(pos)}{M^{\frac{2i}{d}}}\right). \tag{6.4}$$

The word position *pos* is replaced with the level index of the word in the dependency parse tree. The dependency parse trees are obtained using the BOUN dependency parser as in Section 6.1.2.

The $level_index(pos)$ denotes the level index of the word in position *pos* in the dependency parse tree of the sentence.

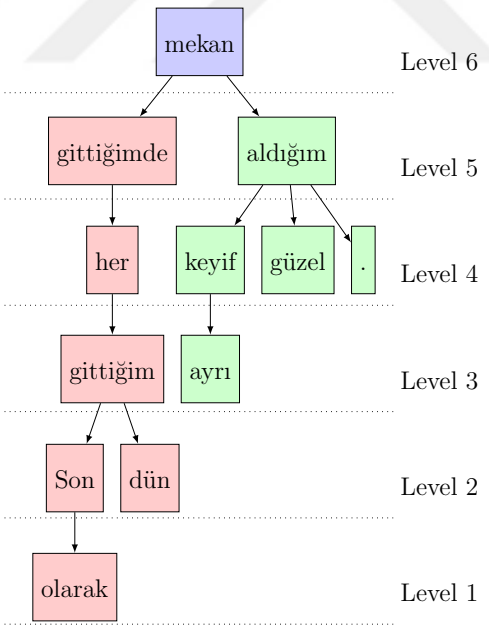


Figure 6.2. Dependency parse tree of the sentence "son olarak dün gittiğim her gittiğimde ayrı keyif aldığım güzel mekan .".

6.1.4. BiLSTM And CRF Layers

The word embedding, the POS tag embedding, and the tree positional encoding of each word in the sentence are concatenated and fed as input to the higher layers of the network. We build the network as a combination of BiLSTM and CRF models in order to effectively capture and label the aspects in the reviews. The BiLSTM layer processes the input words sequentially while capturing the contextual information and dependencies between the words. The LSTM cells maintain a hidden state and a cell state that evolve as new words are processed. The hidden state encodes contextual information from the previous words in the sequence, whereas the cell state holds information about long range dependencies.

The CRF layer takes the LSTM-generated features as input and assigns aspect labels to each word in the sentence by taking into account the relationships between neighboring words. To label the aspects, we use the BIO scheme [48] which is a commonly used labeling scheme in sequence classification tasks. "B" marks the first term of an aspect, "I" marks the intermediate terms (terms except the first term) of the aspect, and "O" marks the nonaspect terms. The reason for using a CRF model on top of the LSTM model is that the CRF model considers the label transitions between the words and aims to find the optimal sequence of labels for the entire sentence. For each word in the sentence, the CRF produces scores for the three aspect labels that reflect the compatibility of each label with the context. In addition, the transition probabilities from the aspect labels in a word position to the labels in the next position are kept in a transition matrix.

We employ an end-to-end training scheme in the sense that the error computed as the output of the CRF layer for each word is backpropagated through the entire network until the input embeddings and the parameters are updated. In the first model where random or Word2Vec embeddings are used as the initial input embeddings, the training scheme involves updating the initial embeddings.

In the second model where the tokens are fed to the BERT model to produce the word embeddings, the training scheme indicates that the token embeddings as well as the parameters of the BERT network are updated. As a loss function, we use the negative log-likelihood loss which measures the difference between the predicted aspect label probabilities and true label probabilities. In this manner, the integrated model is trained to optimize a global objective function that takes into account the contributions of all the BERT, BiLSTM, and CRF components to the overall aspect extraction task.

6.1.5. Viterbi Decoding

During the test phase of the proposed models, we use the Viterbi decoding algorithm to obtain the aspect labels of the words in the sentence as output. The Viterbi algorithm [142] is a dynamic programming algorithm commonly used to find the most likely sequence of hidden states in a hidden Markov model or CRF model. In the context of sequence labeling tasks similar to aspect extraction, the algorithm determines the optimal sequence of labels for a given input sequence by considering the transition probabilities between the labels and the likelihood of the words. Beginning with the first word in the sentence, the algorithm iteratively selects the best aspect label for each word by considering the transition and likelihood probabilities.

6.1.6. Evaluation

We evaluate the results in terms of F1-values. 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. The F1-value of a class is calculated using

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{6.5}$$

where TP, FP, and FN correspond to true positives, false positives, and false negatives, respectively.

6.2. Experiments And Results

In this section, we explain the experiments conducted with the proposed models and give the performance of the models for Turkish aspect extraction. We consider the two models stated in Section 6.1 separately. We analyze their performance in detail with different input parameters and different initialization conditions of the parameters. We give the results on the standard train/test splits of the datasets in order to be able to compare the results with those in the literature. In addition, we also give the results obtained with k-fold cross-validation (k=5) to show the generalization performance of the models.

Tables 6.1 and 6.2 show the results on the SemEval 2016 Task 5 Restaurant Turkish Reviews dataset. Table 6.1 depicts the result of the first model where random or Word2Vec embeddings are used for the word and POS tag embeddings. The first column denotes which inputs (words, POS tags, position information) are used (+) in the experiment. The second column shows how the word and POS tag embeddings are initialized as random vectors or as Word2Vec vectors. The next column specifies the type of positional information used, where *positional* stands for the positional encoding used in the original transformer model and TPE is the tree positional encoding proposed in this work. The last two columns are the results with the standard train/test splits of the datasets and with k-fold cross-validation.

We see that using random vectors or Word2Vec vectors to initialize the word and POS tag embeddings does not cause a consistent change in the results for both of the splits. Random initialization outperforms Word2Vec initialization in some cases while Word2Vec vectors show a better behavior in the others. This may be attributed to both types of the vectors being adapted to the task domain during the training phase of the aspect extraction models. Another observation is that the tree positional encodings show better performance than the original encodings. This result signals that using the distance of the words in a sentence from the head of the sentence provides useful information in identifying the aspects in the sentence.

Table 6.1. BiLSTM-CRF model macro F1-values for Turkish SemEval 2016 restaurant reviews dataset.

Input			Word/POS Embedding	Position Encoding	Macro F-1 for Original Train-Test Split	Macro F-1 for K-Fold
Word	**POS Tag**	**Position**				
+	-	-	Random	-	46.39	47.67
+	-	-	Word2Vec	-	50.63	49.98
+	+	-	Random	-	54.65	49.63
+	+	-	Word2vec	-	52.50	52.13
+	-	+	Random	Positional	47.95	49.25
+	-	+	Random	TPE	49.41	49.26
+	-	+	Word2vec	TPE	48.45	46.66
+	+	+	Random	TPE	48.77	51.41
+	+	+	Word2vec	TPE	48.61	53.91

Table 6.2 shows the results of the second model where the word embeddings are obtained with the BERT model. We observe a significant increase in the success rates compared to the model with random/Word2Vec word vectors. This result aligns with the literature in the sense that contextualized BERT embeddings outperform non-contextualized embeddings in downstream NLP tasks [2].

Table 6.2. BERT-BiLSTM-CRF model macro F1-values for Turkish SemEval 2016 restaurant reviews dataset.

Input		POS Embedding	Position Encoding	Macro F-1 for Original Train-Test Split	Macro F-1 for K-Fold
POS Tag	**Position**				
-	-	-	-	74.02	66.74
+	-	Random	-	74.13	67.53
+	-	Word2vec	-	74.50	68.36
-	+	-	TPE	68.89	66.89
+	+	Random	TPE	74.88	68.34
+	+	Word2vec	TPE	75.74	68.73

The table also shows that the positional encodings mostly contribute to the performance of the model with different types of POS tag embeddings. When we consider all the results on this dataset shown in Tables 6.1 and 6.2, we see that using BERT embeddings for the words and integrating tree positional encoding information to the network yield the best results with both the standard train/test split and k-fold cross-validation.

Table 6.3. BiLSTM-CRF model macro F1-values for translated English SemEval 2016 restaurant reviews dataset.

Input			**Word/POS Embedding**	**Position Encoding**	**Macro F-1 for Original Train-Test Split**	**Macro F-1 for K-Fold**
Word	**POS Tag**	**Position**				
+	-	+	Random	TPE	59.87	59.79
+	-	+	Word2Vec	TPE	58.88	57.61
+	-	-	Random	-	55.96	59.66
+	-	-	Word2Vec	-	61.33	61.69
+	+	-	Random	-	60.62	61.05
+	+	-	Word2vec	-	61.89	63.89
+	-	+	Random	Positional	60.76	32.47
+	-	+	Random	TPE	55.96	59.79
+	-	+	Word2vec	TPE	58.88	56.33
+	+	+	Random	TPE	59.90	62.12
+	+	+	Word2vec	TPE	58.56	58.19

Tables 6.3 and 6.4 show the results on the machine-translated dataset obtained from the SemEval 2016 Task 5 Restaurant English Reviews dataset. We see a similar behavior as in the other dataset. Using random vectors or Word2Vec vectors for word and POS tag embeddings does not make a significant difference in the results. Including POS information in the models consistently increases the performance compared to using only the word information. In the BERT-based model, tree positional encodings contribute to the performance with different types of POS tag embeddings and also when POS tag embeddings are not used.

As in the previous dataset, the best results in this dataset are obtained with contextualized BERT embeddings. Using tree positional encoding yields the best performance with k-fold cross-validation, whereas it slightly decreases the performance on the standard train/test split.

Table 6.4. BERT-BiLSTM-CRF model macro F1-values for translated English SemEval 2016 restaurant reviews dataset.

Input		**POS Embedding**	**Position Encoding**	**Macro F-1 for Original Train-Test Split**	**Macro F-1 for K-Fold**
POS Tag	**Position**				
-	-	-	-	70.04	65.96
+	-	Random	-	71.89	72.59
+	-	Word2vec	-	71.29	73.07
-	+	-	TPE	70.69	72.23
+	+	Random	TPE	71.07	73.26
+	+	Word2vec	TPE	71.51	73.36

The best results we obtained on the standard train/test splits for the SemEval 2016 Task 5 Restaurant Turkish Reviews dataset and the machine-translated SemEval 2016 Task 5 Restaurant English Reviews dataset are, respectively, 75.74 and 71.89 F1 values. We share the results of the SemEval 2016 Task 5 competition for Turkish and English in Tables 6.5 and 6.6, respectively, and compare with the results obtained in this work. We see that the result of the proposed models used in this work outperforms the shared task results with a large margin on the original Turkish dataset. For the translated Turkish dataset, the result we obtained slightly falls behind the shared task results. This can be attributed to the translated nature of the dataset. The translations are performed in a controlled manner and post-processing steps are applied, however the automatic translation process brings some noise into the texts. Although we used translated reviews in the experiments for this dataset, the results are close to the top scores obtained in the shared task.

Table 6.5. Comparison of models for Turkish SemEval 2016 restaurant reviews dataset.

Model	**F1-value**
Our Model	75.74
UFAL [143]	61.03
Wang et al. [144]	59.30
IIT-T [81]	56.63
INSIG [145]	49.12

Table 6.6. Comparison of models for English SemEval 2016 restaurant reviews dataset.

Model	**F1-value**	**Language**
Wang et al. [146]	81.30	English
Wei et al. [147]	78.00	English
Xu et al. [148]	77.70	English
Xu et al. [149]	75.40	English
Wang et al. [144]	72.80	English
Our Model	71.89	Turkish (Translated)

6.3. Sentiment Analysis

After aspect extraction, we also developed a model to learn the sentiments of the aspects. For that purpose, we got the last layer vectors of the aspect learning model for every token. Then, we calculated the attention vectors of all reviews. For any review, we extracted the attention values of aspect tokens, obtained the attention values of non-aspect tokens, and multiplied embedding vectors of non-aspect tokens with their attention values. We used these attention-weighted embedding vectors as input for a feed-forward linear network with two hidden layers. The algorithm is shown in Figure 6.3 and Table 6.7 shows the results of the sentiment classification for extracted aspects.

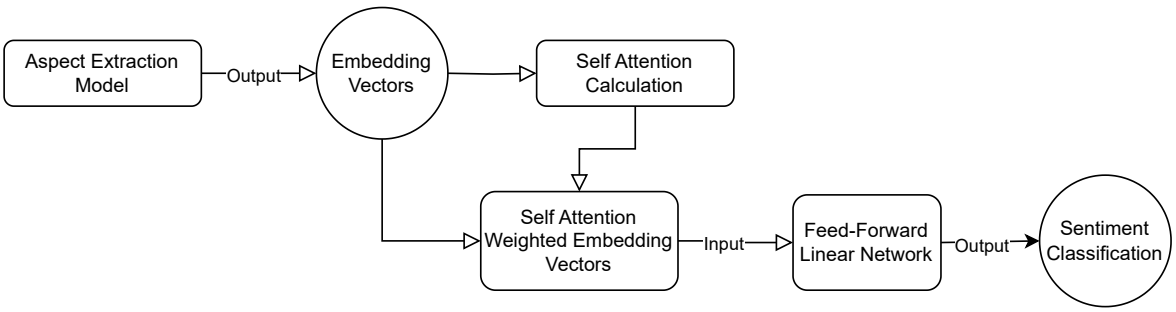


Figure 6.3. Sentiment classification for extracted aspects.

Table 6.7. Sentiment classification for extracted aspects.

Dataset	**Accuracies**
Semeval 2016 Turkish Restaurant Reviews	62.16
Semeval 2016 Translated Restaurant Reviews	68.29

6.4. Conclusion

In this study, we focused on aspect-based sentiment analysis (ABSA), which focuses on aspect extraction by integrating advanced techniques such as BERT, BiLSTM, CRF, embedding, and encoding.

The integration of BERT with its ability to capture contextualized word representations has significantly contributed to the model's ability to distinguish complex relationships between words in a given context. The application of BiLSTM and Tree Positional Encoding (TPE) proved effective in addressing aspects' sequential and hierarchical structure, allowing the model to grasp long-term dependencies critical for accurate extraction. Including CRF as the final layer increased the precision of our model by accounting for sequential dependencies and imposing constraints between adjacent direction predictions. This comprehensive approach increased the model's accuracy in appearance extraction.

Our model has been rigorously evaluated on comparative data sets, including restaurant reviews from customers.

One of these data sets is English and the other is Turkish. We translated the English into Turkish and relabeled the translated dataset. In this way, we have obtained an extensive and labeled review dataset in Turkish. Therefore, we have provided a new dataset for future academic studies.

The results consistently demonstrated superior performance compared to existing methodologies. The combination of BERT, BiLSTM, CRF, and embedding showcased a synergistic effect, underscoring the importance of combining state-of-the-art techniques to address the challenges posed by real-world textual data. We combined the pre-trained Turkish BERT model with BiLSTM and CRF. In addition to word embedding, we add POS tags embedding and our novel tree-positional encoding to the model. In our experiments, we split the datasets for K-fold cross-validation and perform tests in addition to the data sets' original train/test separation.

As we conclude this research, it is evident that the proposed model represents a significant step forward in the field of ABSA. The insights gained from our study contribute to the refinement of aspect extraction techniques and shed light on the intricate interplay between BERT, BiLSTM, CRF, and embedding in unraveling the complex tapestry of sentiments within natural language and provide a new labeled dataset in Turkish for ABSA.

7. CONCLUSION

7.1. Summary of Contributions

The focus of this thesis was on aspect-based sentiment analysis. ABSA consists of these subtasks; sentiment analysis, aspect extraction, aspect classification and opinion extraction. First, we focused on the sentiment analysis subtask. We developed different models with different embeddings. Then we studied aspect classification.

In Chapter 4, we used a combination of Word2vec and GloVe vectors. These vectors were used directly for the CNN model to create review matrices. For our averaged vector model, for a review, we found averaged vectors from word vectors of the word in the review. For our weighted averaged vector model, we first multiplied the word vectors by predefined values, such as the ratio of numbers of positive and negative words. We used these vectors as input to the learning algorithms. We are very close to the state-of-the-art results for the IMDB dataset and the Semeval-2016 Task-5 dataset although we did not use any pre-trained model such as BERT [128] or the Large Language Model LLAMA [133]. Our weighted averaged vector model obtained high accuracy values for the IMDB dataset and is computationally very efficient.

In Chapter 5, 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 a 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). 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 carried out 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 embedding types strongly affect the performance of the models.

In Chapter 6, we focused on aspect extraction by integrating advanced techniques such as BERT, BiLSTM, CRF, embedding and encoding. The integration of BERT with its ability to capture contextualized word representations has significantly contributed to the model's ability to distinguish complex relationships between words in a given context. The application of BiLSTM and novel Tree Positional Encoding (TPE) proved effective in addressing aspects' sequential and hierarchical structure, allowing the model to grasp long-term dependencies critical for accurate extraction. Including CRF as the final layer increased the precision of our model by accounting for sequential dependencies and imposing constraints between adjacent direction predictions. This comprehensive approach increased the accuracy of the model in aspect extraction. Our model has been rigorously evaluated on comparative data sets, including restaurant reviews from customers. One of these data sets is English, and the other is Turkish. We translated the English into Turkish and relabeled the translated dataset. In this way, we have obtained an extensive and labeled review dataset in Turkish. Therefore, we have provided a new dataset for future academic studies. The results consistently demonstrated superior performance compared to the existing methodologies.

As we conclude this research, it is evident that the proposed model represents a significant step forward in the field of ABSA. The insights gained from our study contribute to the refinement of aspect extraction techniques and embedding in unraveling the complex tapestry of sentiments within natural language and provide a new labeled dataset in Turkish for ABSA.

7.2. Future Work

After the development of contextualized embeddings such as BERT, large language models developed by increasing the number of parameters and volume of data also offer great opportunities for ABSA. Here are some potential directions for future research on aspect extraction with large language models:

- By investigating and developing more effective fine-tuning strategies for large language models in the context of sentiment analysis and aspect extraction, we explore novel techniques to enhance the model's ability to identify sentiment and extract specific aspects from diverse domains.
- Large language models can potentially benefit from combining textual information with visual cues, especially in scenarios where aspects are contextually enriched through multimedia content. Therefore, we explore the integration of multimodal data like text or images for aspect extraction.
- We develop techniques that allow large language models to adapt quickly and effectively to new domains with limited labeled data.
- We extend sentiment classification and aspect extraction capabilities to multiple languages. We explore methods for training and fine-tuning large language models on multilingual datasets to ensure effective sentiment classification and aspect extraction in diverse linguistic contexts.
- We explore interactive learning approaches for ABSA, allowing users to provide feedback and corrections to improve the model's performance. We investigate how user interactions can be leveraged to refine the model over time.
- We extend our study to the identification of opinion and aspect classification, which are other ABSA tasks.

REFERENCES

1. Zhang, W., X. Li, Y. Deng, L. Bing and W. Lam, "A Survey On Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges", *Institute of Electrical and Electronics Engineers Transactions On Knowledge and Data Engineering*, Vol. 35, pp. 11019–11038, 2022.

2. Chauhan, G. S., R. Nahta, Y. K. Meena and D. Gopalani, "Aspect Based Sentiment Analysis Using Deep Learning Approaches: A Survey", *Computer Science Review*, Vol. 49, p. 100576, 2023.

3. Raman, S., V. Gupta, P. Nagrath and K. C. Santosh, "Hate and Aggression Analysis in NLP with Explainable AI", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 36, No. 15, p. 2259036, 2022.

4. Rana, J., L. Gaur and K. C. Santosh, "Classifying Customers' Journey From Online Reviews of Amazon Fresh via Sentiment Analysis and Topic Modelling", *3rd International Conference On Computation, Automation and Knowledge Management*, pp. 1–6, Dubai, United Arab Emirates, 2022.

5. Sammut, C. and G. I. Webb, *Encyclopedia of Machine Learning*, chap. TF-IDF, Springer, New Jersey, USA, 2010.

6. Mikolov, T., K. Chen, G. S. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", ArXiv:1301.3781, 2013.

7. Pennington, J., R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation", *Proceedings of the Conference On Empirical Methods in Natural Language Processing*, pp. 1532–1543, Association for Computational Linguistics, Doha, Qatar, 2014.

8. Devlin, J., M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of

Deep Bidirectional Transformers for Language Understanding", *North American Chapter of the Association for Computational Linguistics*, Vol. 1, pp. 4171–4186, 2019.

9. Joulin, A., E. Grave, P. Bojanowski and M. Douze, "FastText.zip: Compressing Text Classification Models", ArXiv:1612.03651, 2016.

10. Liu, Y., 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", ArXiv:1907.11692, 2019.

11. Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning At Scale", ArXiv:1911.02216, 2020.

12. Radford, A., J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models Are Unsupervised Multitask Learners", Semantic Scholar/CorpusID:160025533, 2019.

13. Brown, T., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, "Language Models Are Few-Shot Learners", *Advances in Neural Information Processing Systems*, pp. 1877–1901, 2020.

14. Goldberg, Y. and O. Levy, "Word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding", ArXiv:1402.3722, 2014.

15. Mikolov, T., I. Sutskever, K. Chen, G. Corrado and J. Deanf, "Distributed Representations of Words and Phrases and their Compositionality", *Advances in Neural*

Information Processing Systems, Vol. 26, pp. 1–9, Curran Associates, Inc., 2013.

16. Wenzek, G., M. A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin and E. Grave, "CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data", ArXiv:1911.00359, 2019.

17. Bird, S., E. Klein and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media, Sebastopol, CA, USA, 2009.

18. Porter, M. F., *Readings in Information Retrieval*, chap. An Algorithm for Suffix Stripping, pp. 313–316, Morgan Kaufmann Publishers, San Francisco, CA USA, 1997.

19. Manning, C. D., M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit", *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, Baltimore, Maryland, 2014.

20. Sak, H., T. Güngör and M. Saraçlar, "Turkish Language Resources: Morphological Parser, Morphological Disambiguator, and Web Corpus", *Advances in Natural Language Processing*, Vol. 5221, pp. 417–427, 2022.

21. Miller, G. A., "WordNet: A Lexical Database for English", *Communications of the Association For Computing Machinery*, Vol. 38, No. 11, pp. 39–41, 1995.

22. Qi, P., Y. Zhang, Y. Zhang, J. Bolton and C. Manning, "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online, 2020.

23. Carstairs-McCarthy, A., *An Introduction To English Morphology*, Edinburgh University Press, Edinburgh, UK, 2002.

24. Öztaner, S. M., "A Word Grammar of Turkish with Morphophonemic Rules", ArXiv:cmp-lg/9608013, 1996.

25. Virpioja, S., P. Smit, S. A. Grönroos and M. Kurimo, *Professor 2.0: Python Implementation and Extensions for Morfessor Baseline*, Aalto University Publication Series, Aalto, Canada, 2013.

26. Sennrich, R., B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units", ArXiv:1508.07909, 2015.

27. Gage, P., "A New Algorithm for Data Compression", *The C Users Journal*, Vol. 12, No. 2, pp. 23–38, 1994.

28. Kudo, T. and J. Richardson, "SentencePiece: A Simple and Language-Independent Subword Tokenizer and Detokenizer for Neural Text Processing", *Proceedings of the 2018 Conference On Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, 2018.

29. Schuster, M. and K. Nakajima, "Japanese and Korean voice search", *International Conference on Acoustics, Speech and Signal Processing*, pp. 5149–5152, Kyoto, Japan, 2012.

30. Kudo, T., "Subword Regularization: Improving Neural Network Translation Models with Multiple", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 66–75, Melbourne, Australia, 2018.

31. Büyükkuşcu, I. and E. Adalı, "Heceleme Yöntemiyle Kök Sözcük Üretme", *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, Vol. 2, No. 1, pp. 66–71, 2016.

32. Aydın, C. R. and T. Güngör, "Sentiment Analysis in Turkish: Supervised, Semi-Supervised, and Unsupervised Techniques", *Natural Language Engineering*, Vol. 27, No. 4, pp. 455–483, 2021.

33. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, p. 1735–80, 1997.

34. Lafferty, J. D., A. McCallum and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *In Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289, San Francisco, CA, USA, 2001.

35. Aydın, C. R., T. Güngör and A. Erkan, "Generating Word and Document Embeddings for Sentiment Analysis", *20th International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 307–318, Springer, Switzerland, 2019.

36. Erkan, A. and T. Güngör, "Sentiment Analysis Using Averaged Weighted Word Vector Features", *Public Library of Science One*, Vol. 19, No. 4, p. e0299264, 2024. Reprinted, with permission.

37. Erkan, A. and T. Güngör, "Analysis of Deep Learning Model Combinations and Tokenization Approaches in Sentiment Classification", *Institute of Electrical and Electronics Engineers Access*, Vol. 11, pp. 134951–134968, © 2023 Institute of Electrical and Electronics Engineers. Reprinted, with permission.

38. Erkan, A. and T. Güngör, "Aspect Extraction From Turkish Reviews By Using BERT, BiLSTM, CRF, and Tree Positional Encoding", It has been submitted into Expert Systems with Applications.

39. Chevalier, J. A. and D. Mayzlin, "The Effect of Word of Mouth on Sales: Online Book Reviews", *Journal of Marketing Research*, Vol. 43, No. 3, pp. 345–354, 2006.

40. Liu, B., *Sentiment Analysis and Opinion Mining*, Morgan and Claypool, Kentfield, Canada, 2012.

41. Cambriai, E., B. W. Schuller, Y. Xia and C. Havasi, "New Avenues in Opinion

Mining and Sentiment Analysis", *IEEE Intelligent Systems*, Vol. 28, No. 2, pp. 15–21, 2013.

42. Seki, Y., L. Ku, L. Sun, H. H. Chen and N. Kando, "Overview of Multilingual Opinion Analysis Task At NTCIR-8: A Step Toward Cross Lingual Opinion Analysis", *In Proceedings of the 8th National Institute of Informatics Research Workshop*, pp. 209–220, Tokyo, Japan, 2010.

43. Mitchell, M., "Overview of the TAC2013 Knowledge Base Population Evaluation English Sentiment Slot Filling", *In Proceedings of the 6th Text Analysis Conference*, Gaithersburg, Maryland, USA, 2013.

44. Nakov, P., S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter and T. Wilson, "SemEval-2013 Task 2: Sentiment Analysis in Twitter", *In Proceedings of the 7th International Workshop On Semantic Evaluation*, Atlanta, Georgia, 2013.

45. Rosenthal, S., A. Ritter, P. Nakov and V. Stoyanov, "SemEval-2014 Task 4: Sentiment Analysis in Twitter", *In Proceedings of the 8th International Workshop On Semantic Evaluation*, Dublin, Ireland, 2014.

46. Pontiki, M., D. Galanis, J. Pavlopoulos, S. Manandhar and I. Androutsopoulos, "SemEval-2015 Task 12: Aspect Based Sentiment Analysis", *In Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, Colorado, 2015.

47. Mohammad, S. M., S. Kiritchenko, P. Sobhani, X. Zhu and C. Cherry, "SemEval-2016 Task 6: Detecting Stance in Tweets", *In Proceedings of the 10th International Workshop on Semantic Evaluation*, San Diego, California, 2016.

48. Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. D. Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra and G. Eryiğit, "SemEval-2016 Task 5: Aspect Based Sentiment Analysis", *Pro-*

ceedings of the 10th International Workshop on Semantic Evaluation, pp. 19–30, San Diego, California, 2016.

49. Recupero, D. R. and E. Cambria, "Eswc'14 Challenge on Concept-Level Sentiment Analysis", *In Semantic Web Evaluation Challenge*, pp. 3–20, Anissaras, Crete, Greece, 2014.

50. Ruppenhofer, J., R. Klinger, J. M. Struß, J. Sonntag and M. Wiegand, "IGGSA Shared Tasks on German Sentiment Analysis (GESTALT)", *In Workshop Proceedings of the 12th Edition of the Konferenz zur Verarbeitung natürlicher Sprache/Conference on Natural Language Processing Conference*, p. 164–173, Hildesheim, Germany, 2014.

51. Loukachevitch, N., P. Blinov, E. Kotelnikov, Y. Rubtsova, V. Ivanov and E. Tutubalina, "SentiRuEval: Testing Object-Oriented Sentiment Analysis Systems in Russian", *In Proceedings of International Conference Dialog*, Moscow, RGGU, 2015.

52. Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng and C. Potts, "Recursive Deep Models for Semantic Compositionality over A Sentiment Treebank", *In Proceedings of Empirical Methods in Natural Language Processing*, pp. 1631–1642, Stroudsburg, PA, 2013.

53. Ganu, G., N. Elhadad and A. Marian, "Beyond the Stars: Improving Rating Predictions Using Review Text Content", *In Proceedings of International Workshop on the Web and Databases*, 2009.

54. Klinger, R. and P. Cimiano, "The USAGE Review Corpus for Fine-Grained Multi-Lingual Opinion Analysis", *In Proceedings of the Ninth International Conference On Language Resources and Evaluation*, Reykjavik, Iceland, 2014.

55. Jiménez-Zafra, S. M., G. Berardi, A. Esuli, D. Marcheggiani, M. T. Martín-

Valdivia and A. M. Fernández, "A Multilingual Annotated Dataset for Aspect-Oriented Opinion Mining", *In Proceedings of Empirical Methods in Natural Language Processing*, p. 2533–2538, Lisbon, Portugal, 2015.

56. Lambert, P., "Aspect-Level Cross-lingual Sentiment Classification with Constrained SMT", *In Proceedings of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pp. 781–787, Beijing, China, 2015.

57. Maas, A. L., R. E. Daly, T. P. Peter, D. Huang, A. Y. Ng and C. Potts, "Learning Word Vectors for Sentiment Analysis", *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 150–152, Portland, Oregon, USA, 2011.

58. "Yelp Dataset Challenge", 2014, `https://www.yelp.com/dataset/challenge`, accessed on November 21, 2017.

59. Mesnil, G., T. Mikolov, M. A. Ranzato and Y. Bengio, "Ensemble of Generative and Discriminative Techniques for Sentiment Analysis of Movie Reviews", *International Conference on Learning Representation*, San Diego, CA, USA, 2015.

60. Wang, S. and C. D. Manning, "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics Short Papers*, Vol. 2, pp. 90–94, Jeju Island, Korea, 2012.

61. Hu, M. and B. Liu, "Mining and Summarizing Customer Reviews", *In Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining.*, pp. 168–177, Seattle, WA, USA, 2004.

62. G., Q., B. Liu, J. Bu and C. Chen, "Opinion Word Expansion and Target Extraction Through Double Propagation", *Computational Linguistics*, Vol. 37, No. 1, p.

9–27, 2011.

63. Cortes, C. and V. Vapnik, *Machine learning*, Vol. 20, chap. Support-Vector Networks, pp. 273–297, Springer, New York, USA, 1995.

64. Mucherino, A., P. J. Papajorgji and P. M. Pardalos, *Data Mining in Agriculture*, chap. k-Nearest Neighbor Classification, pp. 83–106, Springer, New York, USA, 2009.

65. Baum, L. E. and T. Petrie, *The Annals of Mathematical Statistics*, Vol. 37, chap. Statistical Inference for Probabilistic Functions of Finite State Markov Chains, pp. 1554–1563, Durham, NC, USA, 1996.

66. Webb, G. I., *Encyclopedia of Machine Learning*, chap. Naïve Bayes, pp. 713–714, Springer, New York, USA, 2010.

67. Kamal, N., "Using Maximum Entropy for Text Classification", *In International Joint Conferences on Artificial Intelligence, Workshop on Machine Learning for Information Filtering*, pp. 61–67, Stockholm, Sweden, 1999.

68. Nasim, Z., Q. Rajput and S. Haider, "Sentiment Analysis of Student Feedback Using Machine Learning and Lexicon-Based Approaches", *International Conference on Research and Innovation in Information Systems*, pp. 1–6, Langkawi, Malaysia, 2017.

69. Dang, N. C., M. N. Moreno-García and F. D. L. Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study", *Electronics*, Vol. 9, No. 3, pp. 483–512, 2020.

70. Kastrati, Z., L. Ahmedi, A. Kurti and F. Gashi, "A Deep Learning Sentiment Analyser for Social Media Comments in Low-Resource Languages", *Electronics*, Vol. 10, pp. 1133–1152, 2021.

71. Chandra, R. and A. Krishna, "COVID-19 Sentiment Analysis via Deep Learning During the Rise of Novel Cases", *Public Library of Science One*, Vol. 16, No. 8, p. e0255615, 2021.

72. Hung, B. T., "Domain-Specific vs. General-Purpose Word Representations in Sentiment Analysis for Deep Learning Models", *Frontiers in Intelligent Computing: Theory and Applications*, Vol. 1013, pp. 252–264, 2020.

73. Shin, B., T. Lee and J. D. Choi, "Lexicon Integrated CNN Models with Attention for Sentiment Analysis", ArXiv:1610.06272, 2016.

74. Camacho-Collados, J. and M. T. Pilevar, "On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis", ArXiv:1707.01780, 2017.

75. Haonan, L., H. H. Seth, Y. Tian and X. Guo, "Graph Star Net for Generalized Multi-Task Learning", ArXiv:1906.12330, 2019.

76. Wang, S., H. Fang, M. Khabsa, H. Mao and H. Ma, "Entailment As Few-Shot Learner", ArXiv:2104.14690, 2021.

77. Chi, S., Q. Xipeng, X. Yige and H. Xuanjing, "How To Fine-Tune BERT for Text Classification", ArXiv:1905.05583, 2019.

78. Qizhe, X., D. Zihang, H. H. Eduard, L. Minh-Thang and V. Quoc, "Unsupervised Data Augmentation for Consistency Training", ArXiv:1904.12848, 2019.

79. Khalil, T. and S. R. El-Beltagy, "NileTMRG At SemEval-2016 Task 5: Deep Convolutional Neural Networks for Aspect Category and Sentiment Extraction", *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 271–276, San Diego, California, 2016.

80. Yoon, K., "Convolutional Neural Networks for Sentence Classification", *Proceed-*

ings of the Conference on Empirical Methods in Natural Language Processing), pp. 1746–1751, 2014.

81. Kumar, A., 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", *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 1129–1135, San Diego, California, 2016.

82. Brun, C., J. Perez and C. Roux, "XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modeling On Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis", *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 277–281, San Diego, California, 2016.

83. Ait-Mokhtar, S., J. P. Chanod and C. Roux, "Robustness Beyond Shallowness: Incremental Deep Parsing", *Natural Language Engineering*, Vol. 8, No. 2-3, pp. 121–144, 2002.

84. Hansen, P. C., "The Truncated SVD As A Method for Regularization", Technical report, Stanford University, 1986.

85. Zou, H. and T. Hastie, "Regularization and Variable Selection Via the Elastic Net", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 67, No. 2, pp. 301–320, 2003.

86. Wallaart, O. and F. Frasincar, "A Hybrid Approach for Aspect-Based Sentiment Analysis Using A Lexicalized Domain Ontology and Attentional Neural Models", *European Semantic Web Symposium*, pp. 363–378, Portoroz, Slovenia, 2019.

87. Shiliang, Z. and R. Xia, "Left-Center-Right Separated Neural Network for Aspect-Based Sentiment Analysis with Rotatory Attention", ArXiv:1802.00892, 2018.

88. Trusca, M. M., D. Wassenberg, F. Frasincar and R. Dekker, "A Hybrid Approach

for Aspect-Based Sentiment Analysis Using Deep Contextual Word Embeddings and Hierarchical Attention", ArXiv:2004.08673, 2020.

89. Reddy, N., P. Singh and M. M. Srivastava, "Does BERT Understand Sentiment? Leveraging Comparisons Between Contextual and Non-Contextual Embeddings To Improve Aspect-Based Sentiment Models", ArXiv:2011.11673, 2020.

90. Cliche, M., "BB_twtr At SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs", *Proceedings of the 11th International Workshop on Semantic Evaluation*, pp. 573–580, Vancouver, Canada, 2017.

91. Yichun Yin, Y. S. and M. Zhang, "NNEMBs At SemEval-2017 Task 4: Neural Twitter Sentiment Classification: A Simple Ensemble Method with Different Embeddings", *Proceedings of the 11th International Workshop on Semantic Evaluation*, pp. 621–625, Vancouver, Canada, 2017.

92. Lei, T., H. Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, A. Moschitti and M. Lluís, "Semi-Supervised Question Retrieval with Gated Convolutions", *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1279–1289, 2016.

93. Hamdan, H., "Senti17 At SemEval-2017 Task 4: Ten Convolutional Neural Network Voters for Tweet Polarity Classification", ArXiv:1705.02023, 2017.

94. Christos Baziotis, N. P. and C. Doulkeridis, "DataStories At SemEval-2017 Task 4: Deep LSTM with Attention for Message-Level and Topic-Based Sentiment Analysis", *Proceedings of the 11th International Workshop on Semantic Evaluation*, pp. 747–754, Vancouver, Canada, 2017.

95. Sebastian Ruder, P. G. and J. G. Breslin, "INSIGHT-1 At SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-Based Sentiment Analysis", *Proceedings*

of the 10th International Workshop on Semantic Evaluation, pp. 330–336, San Diego, California, 2016.

96. Alaettin Uçan, E. A. S., Behzad Naderalvojoud and H. Sever, "SentiWordNet for New Language: Automatic Translation Approach", *12th International Conference on Signal-Image Technology & Internet-Based Systems*, pp. 308–315, Naples, Italy, 2016.

97. Aydın, C. R., T. Güngör and A. Erkan, "Generating Word and Document Embeddings for Sentiment Analysis", *20th International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 1–6, La Rochelle, France, 2019.

98. Turney, P. D., "Thumbs Up Or Thumbs Down? Semantic Orientation Applied To Unsupervised Classification of Reviews", *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424, Philadelphia, Pennsylvania, USA, 2002.

99. Kaya, M., G. Fidan and I. H. Toroslu, "Sentiment Analysis of Turkish Political News", *In Proceedings of the International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, Vol. 1, pp. 174–180, Macau, China, 2012.

100. Çetin, M. and M. F. Amasyalı, "Active Learning for Turkish Sentiment Analysis", *In Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*, pp. 1–4, Albena, Bulgaria, 2013.

101. Türkmenoğlu, C. and A. C. Tantuğ, "Sentiment Analysis in Turkish Media", *In Proceedings of the Workshop On Issues of Sentiment Discovery and Opinion Mining, International Conference on Machine Learning*, pp. 1–11, New York, NY, USA, 2014.

102. Vural, G., B. Cambazoğlu, O. Z. Tokgöz and P. Karagöz, "A Framework for Sentiment Analysis in Turkish: Application To Polarity Detection of Movie Reviews

in Turkish", *In Proceedings of the 27th International Symposium on Computer and Information Sciences*, pp. 437–445, Paris, France, 2012.

103. Dehkharghani, R., Y. Saygın, B. Yanıkoğlu and K. Oflazer, "SentiTurkNet: A Turkish Polarity Lexicon for Sentiment Analysis", *Language Resources and Evaluation*, Vol. 50, No. 3, pp. 667–685, 2016.

104. Jihan, N., Y. Senarath, D. Tennekoon, M. Wickramarathne and S. Ranathunga, "Multi-Domain Aspect Extraction Using Support Vector Machines", *Conference on Computational Linguistics and Speech Processing*, pp. 308–322, Taipei, Taiwan, 2017.

105. Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning Internal Representations By Error Propagation", Tech. rep. ICS 8504 1985. San Diego, California: Institute for Cognitive Science, University of California.

106. Lecun, Y., L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied To Document Recognition", *In Proceedings of the Institute of Electrical and Electronics Engineers*, Vol. 86, pp. 2278–2324, Phoenix, AZ, USA, 1998.

107. Poria, S., E. Cambria and A. Gelbukh, "Aspect Extraction for Opinion Mining with A Deep Convolutional Neural Network", *Knowledge-Based Systems*, Vol. 108, p. 42–49, 2016.

108. Wang, Y., M. Huang, X. Zhu and L. Zhao, "Attention-Based LSTM for Aspect-Level Sentiment Classification", *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, Austin, Texas, 2016.

109. Cui, B., Y. Li, M. Chen and Z. Zhang, "Fine-Tune BERT with Sparse Self-Attention Mechanism", *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3548–3553, Hong Kong, China, 2019.

110. Chauhan, G. S., Y. K. Meena, D. Gopalani and R. Nahta, "A Two-Step Hybrid Unsupervised Model with Attention Mechanism for Aspect Extraction", *Expert Systems with Applications*, Vol. 161, p. 113673, 2020.

111. Li, Z., X. Xie, H. Li, Z. Xu, Y. Li and Y. Liu, "Cross-Lingual Transfer Learning for Statistical Type Inference", *Proceedings of the 31st International Symposium on Software Testing and Analysis*, pp. 239–250, New York, NY, USA, 2022.

112. Ruder, S., P. Ghaffari and J. Breslin, "Deep Learning for Multilingual Aspect-Based Sentiment Analysis", *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 330–336, San Diego, California, 2016.

113. Rosenthal, S., N. Farra and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter", *Proceedings of the 11th International Workshop on Semantic Evaluation*, pp. 502–518, San Diego, California, 2017.

114. Ettaleba, M., A. Barhoumia, N. Camelina and N. Dugué, "Evaluation of Weakly-Supervised Methods for Aspect Extraction", *26th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 2688–2697, Verona, Italy, 2022.

115. Kanwal, A., I. N. Muhammad, Z. Zhiyun, L. Dun, U. Inam, A. Muhammad, Y. G. Yazeed and G. M. Heba, "Breaking Down Linguistic Complexities: A Structured Approach to Aspect-Based Sentiment Analysis", *Journal of King Saud University*, Vol. 35, No. 8, p. 101651, 2023.

116. Alharbi, M., J. Yin and H. Wang, "Surveying the Landscape: Compound Methods for Aspect-Based Sentiment Analysis", *Databases Theory and Applications: 34th Australasian Database Conference*, pp. 101–115, Melbourne, VIC, Australia, 2023.

117. Xiaoyan, L., R. Raga and S. Xuemei, "GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews", *Hindawi Journal of Sensors*, Vol. 2022, pp. 1687–

725X, 2022.

118. Al-Saqqa, S. and A. Awajan, "The Use of Word2vec Model in Sentiment Analysis: A Survey", *Proceedings of the International Conference on Artificial Intelligence, Robotics and Control*, pp. 39–43, Majorca Island, Spain, 2019.

119. Kim, Y., "Convolutional Neural Networks for Sentence Classification", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014.

120. Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural Language Processing (Almost) From Scratch", *Journal of Machine Learning Research*, Vol. 12, pp. 2493–2537, 2011.

121. Chollet, F., "Keras Convolutional Layer", 2015, `https://github.com/fchollet`, accessed on January 21, 2021.

122. Zeiler, M. D., "ADADELTA: An Adaptive Learning Rate Method", ArXiv:1212.5701, 2012.

123. Chi, S., Q. Xipeng, X. Yige and H. Xuanjing, "How To Fine-Tune BERT for Text Classification", ArXiv:1905.05583, 2019.

124. Qizhe, X., D. Zihang, H. H. Eduard, L. Minh-Thang and V. L. Quoc, "Unsupervised Data Augmentation for Consistency Training", ArXiv:1904.12848, 2019.

125. Wang, S., H. Fang, M. Khabsa, H. Mao and H. Ma, "Entailment As Few-Shot Learner", ArXiv:2104.14690, 2021.

126. Haonan, L., H. Seth, Y. Tian and X. Guo, "Graph Star Net for Generalized Multi-Task Learning", ArXiv:1906.12330, 2019.

127. McNemar, Q., "Note On the Sampling Error of the Difference Between Correlated

Proportions or Percentages", *Psychometrika*, Vol. 12, No. 2, p. 153157, 1947.

128. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, Vol. 1, pp. 4171–4186, Minneapolis, Minnesota, 2019.

129. Reddy, N., P. Singh and M. Srivastava, "Does BERT Understand Sentiment? Leveraging Comparisons Between Contextual and Non-Contextual Embeddings To Improve Aspect-Based Sentiment Models", ArXiv:2011.11673, 2020.

130. Trusca, M., D. Wassenberg, F. Frasincar and R. Dekker, "A Hybrid Approach for Aspect-Based Sentiment Analysis Using Deep Contextual Word Embeddings and Hierarchical Attention", ArXiv:2004.08673, 2020.

131. Klambauer, G., T. Unterthiner, A. Mayr and S. Hochreiter, "Self-Normalizing Neural Networks", ArXiv:1706.02515, 2017.

132. Talaat, A., "Sentiment Analysis Classification System Using Hybrid BERT models", *Journal of Big Data*, Vol. 10, No. 1, p. 110, 2023.

133. Touvron, H., T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave and G. Lample, "LLaMA: Open and Efficient Foundation Language Models", ArXiv:2302.13971, 2023.

134. Katamba, F. and J. Stonham, *Macmillan Modern Linguistics - Morphology*, Red Globe Press, London, UK, 2006.

135. Bird, S., E. Klein and E. Loper, *Natural Language Processing with Python*, O'Reilly Media, Sebastopol, CA, USA, 2009.

136. Jurafsky, D. and J. H. Martin, *Speech and Language Processing: An Introduction*

to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd Edition), Pearson Education, Hoboken, NJ, USA, 2019.

137. Manning, C. D., P. Raghavan and H. Schütze, *Introduction To Information Retrieval*, Cambridge University Press, Cambridge, England, 2008.

138. Wu, Y., M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, S. G. L. Kaiser, Y. Kato, H. K. T. Kudo, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", ArXiv:1609.08144, 2016.

139. Mohammed, A. and R. Kora, "A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges", *Journal of King Saud University - Computer and Information Sciences*, Vol. 35, pp. 757–774, 2023.

140. Özateş, S. B., A. Özgür, T. Güngör and B. Başaran, "A Hybrid Deep Dependency Parsing Approach Enhanced with Rules and Morphology: A Case Study for Turkish", *Institute of Electrical and Electronics Engineers Access*, Vol. 10, pp. 93867–93886, 2022.

141. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser and I. Polosukhin, "Attention is All You Need", *In Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, Red Hook, NY, USA, 2017.

142. Viterbi, A., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Transactions on Information Theory*, Vol. 1967, pp. 260–269, 1967.

143. Tamchyna, A. and K. Veselovská, "UFAL At SemEval-2016 Task 5: Recurrent

Neural Networks for Sentence Classification", *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 367–371, San Diego, California, 2016.

144. Wang, X., Y. Jiang, N. Bach, T. Wang, F. Huang and K. Tu, "Structure-Level Knowledge Distillation for Multilingual Sequence Labeling", *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3317–3330, Online, 2020.

145. Ruder, S., P. Ghaffari and J. Breslin, "INSIGHT-1 At SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis", *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 330–336, San Diego, California, 2016.

146. Wang, X., Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang and K. Tu, "Automated Concatenation of Embeddings for Structured Prediction", *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Vol. 1, pp. 2643–2660, Online, 2021.

147. Wei, Z., Y. Hong, B. Zou, M. Cheng and J. Yao, "Don't Eclipse Your Arts Due To Small Discrepancies: Boundary Repositioning with A Pointer Network for Aspect Extraction", *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3678–3684, Online, 2020.

148. Xu, H., B. Liu, L. Shu and P. Yu, "BERT Post-Training for Review Reading Comprehension and Aspect-Based Sentiment Analysis", *In Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*, Vol. 1, pp. 2324–2335, Minneapolis, Minnesota, 2019.

149. Xu, H., B. Liu, L. Shu and P. Yu, "Double Embeddings and CNN-Based Sequence Labeling for Aspect Extraction", *In Proceedings of the 56th Annual Meeting of*

the Association for Computational Linguistics, Vol. 2, pp. 592–598, Melbourne, Australia, 2018.

APPENDIX A: TELİF HAKLARI

Bu tez çalışması kapsamında ortaya çıkan ve telif hakkı yayınevine devredilen görseller, yayınevinin kendi ağ sayfasında bulunan "yazarın kendi ürettiği yazı ve grafiklerin tekrar kullanımı hakkında geçerli olan yayın politikası"na uygun olarak tez kitabında kullanılmıştır.

APPENDIX B: IEEE LICENSES AND COPYRIGHT

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Boğaziçi University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to IEEE web page [13] to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

[13] http://www.ieee.org/publications_standards/publications/rights/rights_link.html

APPENDIX C: PLOS ONE LICENSES AND COPYRIGHT

PLOS applies the Creative Commons Attribution 4.0 International (CC BY) license [14] , or other comparable licenses that allow free and unrestricted use, to articles and other works we publish. If you submit your paper for publication by PLOS, you agree to have the CC BY license applied to your work. PLOS requires that you as the author agree that anyone can reuse your article content in whole or part for any purpose, for free, even for commercial purposes. These permitted uses include but are not limited to self-archiving by authors of submitted, accepted and published versions of their papers in institutional repositories. Anyone may copy, redistribute, reuse, or modify the content as long as the author and original source are properly cited. This facilitates freedom in reuse and also ensures that PLOS content can be mined without barriers for the needs of research.

[14] https://creativecommons.org/licenses/by/4.0/