TEXT CLASSIFICATION VIA WORD EMBEDDINGS:

AN APPLICATION FOR TURKISH MUSIC MOOD DETECTION

BARIŞ ÇİMEN

BOĞAZİÇİ UNIVERSITY

2017

TEXT CLASSIFICATION VIA WORD EMBEDDINGS:
AN APPLICATION FOR TURKISH MUSIC MOOD DETECTION

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Barış Çimen

Boğaziçi University

2017

DECLARATION OF ORIGINALITY

I, Barış Çimen, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature .....

Date 13.07.2017

ABSTRACT

Text Classification via Word Embeddings:

An Application for Turkish Music Mood Detection

The objective of this study is to bring an approach that incorporates word embeddings into Turkish text classification process, and to evaluate the applicability and performance of this approach by applying it for Turkish music mood detection. The methodology followed in this study consists of two main parts. In the first part, word embeddings are trained through a large collection of textual data, which includes more than 2.5 million Turkish documents gathered from the Internet, by using Word2Vec and GloVe algorithms. Subsequently, lyrics vectors are generated for the pre-processed lyrics selected for mood detection through the use of word embeddings that were trained initially. In the second part of the study, lyrics vectors are employed as features in music mood detection performed via various machine-learning techniques. Besides, Turkish music mood detection is also done by using traditional bag-of-words approach, in which TF-IDF term weighting scheme is used, and Doc2Vec algorithm for comparison purposes. The effects of stemming of the words into their roots and filtering out the precompiled list of stop-words on the results are investigated as well. The results obtained from the study show the effectiveness of incorporating word embeddings generated using big textual data collection into the Turkish text classification process, which is clearly illustrated by the improved classification performance.

ÖZET

Kelime Temsilleri Yoluyla Metin Sınıflaması:
Türkçe Müziklerde Duygu Tespiti Uygulaması

Bu çalışmanın amacı, Türkçe metin tabanlı sınıflandırma işlemine kelime temsillerini dâhil eden bir yaklaşım getirmek ve söz konusu yaklaşımın uygulanabilirliğini ve performansını, Türkçe şarkıların müzik duygu analizinde değerlendirmektir. Bu çalışmada izlenen metot, iki temel aşamadan oluşmaktadır. Birinci aşamada kelime temsilleri, Word2Vec ve GloVe algoritmaları kullanılarak internet ortamından toplanan 2,5 milyondan fazla Türkçe dokümandan oluşan metin tabanlı büyük bir veri kümesi ile eğitilmiştir. Ardından oluşturulan kelime vektörleri vasıtasıyla, duygu tespiti yapılmak üzere seçilen ve ön işlemden geçirilen şarkı sözleri için şarkı sözü vektörleri oluşturulmuştur. Çalışmanın ikinci aşmasında ise, oluşturulan şarkı sözü vektörleri, çeşitli makine öğrenmesi teknikleri vasıtasıyla müzik duygu analizi işleminde kullanılmıştır. Karşılaştırma amacıyla, yaygın olarak kullanılan TF-IDF skorlarına dayanan kelime çantası (bag-of-words) yaklaşımı ve Doc2Vec algoritması da Türkçe müzik duygu analizi için düşünülmüştür. Aynı zamanda kelimeleri köklerine ayrıştırma ve önceden derlenmiş olan etkisiz kelimeleri filtreleme işlemlerinin sonuçlar üzerindeki etkisi de araştırılmıştır. Araştırmadan elde edilen sonuçlar, metin tabanlı büyük veri kümesi aracılığıyla oluşturulan kelime temsillerinin, Türkçe metin sınıflandırma sürecine dâhil edilmesinin etkinliğini ve performansı iyileştirdiğini ortaya koymaktadır.

ACKNOWLEDGEMENTS

This work would not have been possible without the guidance and support of certain individuals who extended their valuable assistance in one way or another.

First and foremost, I would like to extend my deepest gratitude to my supervisor, Assist. Prof. Ahmet Onur Durahim, for his continuous support, valuable guidance, and endless patience. I am gratefully indebted to him for his mentorship, which helped me through the process of researching and writing this thesis.

Besides my advisor, I would also like to thank Prof. V. Aslıhan Nasır and Assist. Prof. Cengiz Örencik for taking time to help me finalize my thesis.

My sincere thanks also go to my professors at Boğaziçi University Management Information Systems Department for their continuous assistance. They have always encouraged me to do better.

I want to thank Fügen Demirtaş and my dear colleagues, for always being there for me whenever I needed.

I must express my profound gratitude to Refika Çimen, my beloved wife, for providing me with unfailing support and continuous encouragement and for steering me in the right direction whenever I needed throughout my years of study.

Last but not the least, I would like to thank my parents and brother for their endless encouragement and support throughout my life.

Finally, I would like to thank everyone who was important to the success of this thesis, as well as expressing my apology that I could not mention them personally.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

As the World Wide Web continues to expand dramatically and the volume of online text increases, the development of automated categorization methods becomes even more important (McCallum, Rosenfeld, Mitchell, & Ng, 1998). According to a research by Turner, Gantz, Reinsel, and Minton (2014), from 2013 to 2020, the growth factor of the unstructured data amount on the Internet will be 10 – which means that the relevant amount will increase from 4.4 zettabytes to 44 zettabytes by doubling in size every two years. It is unpractical and almost impossible to categorize such a big volume of unstructured data manually, which makes it essential to develop a continuous automatic categorization process that can make the data more manageable and reachable. To this end, there is a compelling need for approaches and methods in the field of Text Mining (TM) (Chen & Liu, 2004), which is an important research area. The objective of TM (a.k.a. Intelligent Text Analysis, Knowledge Discovery in Text, or Text Data Mining) is to process unstructured text documents in order to extract valuable data and knowledge (Amancio, et al., 2014). TM is an interdisciplinary field which draws on machine learning (Michie, Spiegelhalter, & Taylor, 1994), computational linguistics, information retrieval and statistics compositely. One of the most commonly-used methods in TM studies is the method of Text Classification (a.k.a. text categorization, or topic spotting). Text classification is a process through which text documents are automatically assigned to one or more predefined categories (Sun &

Lim, 2001). In other words, it evaluates uncategorized data based on their contents and categorizes them.

Text classification is mainly characterized by high dimensionality, which can be employed to generate thousands of features (Yang, Qu, & Liu, 2014). Yet, most of the features generated are either irrelevant or cause poor performance of the classifier; therefore, text classification relies upon the stages of pre-processing and feature selection to select a relevant subset from the entire group of features prior to the evaluation of machine learning algorithms (Kılınç, et al., 2017).

This study introduces an approach which incorporates word embeddings into the Turkish text classification process in order to enhance the success of text classification. Word embeddings can establish semantic and structural relationships among words. One of the most effective features of word embeddings is their ability to transfer inter-word similarities and relations to inter-vector distances while converting words into word vectors in a vector space. As a result of this transfer, it is possible to obtain and use some relations between words through the operations (such as addition, subtraction, distance finding) applied on the words, which are expressed as mathematical vectors in vector algebra.

The primary objective of this study is to introduce an approach that incorporates word embeddings into text classification in Turkish language and to test the accuracy of this approach in Turkish Music Mood Detection application, which has lately become a popular research topic in the field of machine learning. As part of the study, three different people categorized music moods of more than 700 random songs in Turkish language based on their lyrics, and at least two of these three annotators agreed on the same category for the lyrics of 515 Turkish songs.

The study method consists of two main parts. In the first part, word embeddings are trained through Word2Vec and GloVe algorithms based on a huge amount of data, which includes more than 2.5 million Turkish documents gathered from the Internet. This process is followed by the generation of lyrics vectors from pre-processed lyrics designated for mood detection by using word embeddings that have been trained in the first place. Additionally, Turkish music mood detection application is conducted by using Doc2Vec algorithm and bag-of-words approach based on TF-IDF scores for comparison purposes. Finally, stemming of words into their roots and constituents, and filtering out a precompiled list of stop-words on the results are investigated. The second part, on the other hand, is centered on the classification process through lyrics vectors by using different machine learning algorithms employed in data mining. In this part, several machine-learning algorithms are used for music mood detection and their performances are compared and analyzed.

CHAPTER 2

LITERATURE REVIEW

2.1 Text classification

Compared to the myriad of previous studies conducted on text classification in other languages in the literature, there is a rather limited number of text classification researches conducted in Turkish. For instance, as one of the most widely-spoken languages in the world, English features plenty of text classification studies (Read, 2005; Cavnar & Trenkle, 1994). Additionally, we see many examples of text classification researches in various European languages such as German and Spanish (Ciravegna, et al., 2000) and in Asian languages including Chinese and Japanese (Peng, Huang, Schuurmans, & Wang, 2003). Surprisingly, there are interesting researches in the literature on some other languages with different morphological characteristics such as Arabic as well. The study of El-Kourdi, Bensaid, and Rachidi (2004) uses Naïve Bayes (NB) algorithm to automatize Arabic document classification, which reportedly produces an average of 68.78% accuracy. Another study for Arabic text classification is proposed by Shaalan and Oudah (2014) who perform named entity recognition by combining different machine learning algorithms. The success rate of the study for giving accurate results is claimed to exceed 90%. Furthermore, there is a study conducted by Al-Radaideh, Al-Eroud, and Al-Shawakfa (2012) in order to detect spam emails composed in Arabic. Accordingly, they use Graham statistical filter and rule-based filter and thus manage to obtain accurate results by correctly filtering 87% of the messages in the dataset.

In the field of Turkish text classification, on the other hand, one of the studies is conducted by Güran, Akyokuş, Güler, and Gürbüz (2009). As part of the study, they evaluate NB, Complement Naïve Bayes (CNB), Multinomial Naïve Bayes (MNB), C4.5 decision tree (J48) and K-Nearest Neighbor (K-NN) text classification algorithms. Their proposed study basically centers on the N-gram approach, while they perform their experimental studies on documents that are either pre-processed or not. According to the experimental evaluation, using bigram and trigram representations and K-NN algorithm in combination, as well as unigram representations and J48 algorithm together, produces the worst results. Whereas using unigram representations and MNB, bigram representations and CNB, and trigram and NB in combination give the best classification results, which are 95.83%, 93.17%, and 52.83%, respectively.

Another study is carried out by Torunoğlu, Çakirman, Ganiz, Akyokuş, and Gürbüz (2011) to observe the importance of pre-processing steps in Turkish text classification. In the study, they evaluate a variety of pre-processing methods from stop-words filtering to word weighting and four text classification algorithms –NB, MNB, K-NN, and Support Vector Machine (SVM), on several Turkish datasets. According to the experimental results, the pre-processing step do not create the expected impact on Turkish text classification.

Akkus and Cakici (2013) suggest that languages with semantic richness like Turkish could benefit from morphological analysis in text classification studies. They examine contribution of morphological analysis to Turkish text classification. First, they identify stems of the words through Fixed Length Stemmer method and evaluate K-NN, SVM and NB learning algorithms on these stems. The evaluation results on the dataset point out that the use of a simple approximation with the first

five characters that represent documents gives 94.37% f1 score, whereas an expensive morphological analysis achieves 91.87% f1 score.

In another study, Özgür, Güngör, and Gürgen (2004) propose an anti-spam filtering method developed for Turkish in particular and specific to agglutinative languages, consisting of two separate modules – the Learning Module and the Morphology Module. The study is based on both Artificial Neural Network and Bayesian Network algorithms. They claim that they achieve 90% success in finding spam emails in Turkish on the dataset.

In another study, Çataltepe, Turan, and Kesgin (2007) research the effect of stem length derived from words on Turkish text classification. They derive short stems from long stems through various methods, without taking the meaning of the words into account. They compare accuracy rates of classifying vectors weighted through TF-IDF method and obtained from stems that consist of fewer characters. They suggest that Centroid classification method gives better results with shortened stems.

The study of Amasyalı and Beken (2009) focuses on a different approach in text classification. As part of the study, words of a text are assigned into a semantic space. Authors state that they achieve better results by representing words in semantic space, compared with the bag-of-words model. According to their experimental results, use of the Linear Regression classification algorithm gives 93.25% accuracy score.

Amasyalı and Diri (2006) propose an n-gram approach for text classification in Turkish language. Their study focuses on evaluating NB, SVM, J48, and Random Forest (RF) classification algorithms. The study results suggest that bigram based classification algorithms give better results compared with trigram based ones. The

results of classification algorithms show that NB can identify the author of the text more successfully, while SVM performs better in determining both genre of the text and gender of the author. The score of NB in determining the author of the text is 83.3%, whereas the scores of SVM in determining genre of the text and gender of the author are 93.6% and 96.3% respectively.

In their study, Uysal and Gunal (2014) suggest that pre-processing has an important role in text classification. Their dataset is based on emails and news written both in English and Turkish, through which they determine how pre-processing methods affect classification of the text documents. They also research the ways in which tokenization, stop-word removal, lower-case conversion and stemming processes and their various combinations affect the accuracy of SVM classification algorithm. The study finds that some pre-processing methods decrease the accuracy score in classification of text documents, while lower-case conversion and stop-word removal processes improve it. In their study, the maximum achieved Micro-F1 score is 97.13% in Turkish email dataset when the feature size is 200, tokenization is alphabetic, stop-word removal is off, lowercase conversion is on, and stemming is off, whereas the maximum achieved Micro-F1 score is 80.61% in Turkish news dataset when the feature size is 2000, tokenization is alphabetic, stop-word removal is off, lowercase conversion is on, and stemming is on.

Gunal (2012) studies the effects of various feature selection approaches on text classification. His studies are based on a hybrid selection method created through the combination of filter and wrapper feature selection methods. Accordingly, features obtained through this method give more successful results in Turkish text classification, as opposed to the single selection method.

A study conducted by Alparslan, Karahoca, and Bahsi (2011) aims to extract information from documents that are classified within the Turkish language. First, word stems are extracted through the use of stemming algorithms which are particularly employed for Turkish text documents. Then, they form document term matrices through the TF-IDF weighting method with stems obtained after preprocessing. Unlike other studies, they combine SVM and Adaptive Neuro Fuzzy classification algorithms in this study. According to the experimental results, their method achieves 96.67% accuracy score.

2.2 Word embeddings

The leading two methods for generating word embeddings are Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), and GloVe (Pennington, Socher, & Manning, 2014). Mikolov et at. (2013), propose skip-gram and continuous bag-of-words (CBOW) models, collectively known as Word2Vec model, where a single-layer neural network architecture is used for learning useful continuous word representations. In the skip-gram model, one word is used to predict the surrounding context words, whereas in CBOW model, a word is predicted in consideration of the surrounding context words. In their proposed model, each word has two vectors and the architecture is based on the inner product of these two word vectors. The proposed approach is a shallow window-based method, where word representations are learned by taking local context windows of each word into account. Essentially, the skip-gram model attempts to optimize a neighborhood-preserving objective in light of the distributional hypothesis which proposes that words in comparable contexts usually represent similar meanings (Harris, 1954). This objective is optimized using stochastic gradient descent with

negative sampling in the study by Mikolov et al. (2013) and with hierarchical softmax in the study by Mikolov, Chen, Corrado, and Dean (2013).

The Word2Vec model suffers from ignoring the global word co-occurrence statistics of a given corpus, and it only considers the context windows of the words across the entire corpus (Pennington, Socher, & Manning, 2014).

In order to address this problem, Pennington et al. (2014), proposed GloVe model, which is also an unsupervised method for learning continuous word vectors similar to the Word2Vec model. But, GloVe differs from the Word2Vec method in that it takes into account the global statistics of word co-occurrences in a given corpus in addition to the statistics of local context windows.

Briefly, both of these methods can establish semantic and structural relationships among words. One of the most powerful features of these methods is that they are able to transfer the similarities and relations between words to the distance between vectors while converting words to word vectors in a vector space. As a result of this transfer, it is possible to obtain and use some relations between words by using the operations used in vector algebra (such as addition, subtraction, distance finding) on the words expressed as mathematical vectors.

In a study by Su, H. Xu, Zhang, and Y. Xu (2014), Word2Vec and SVM are used in combination in the classification of Chinese comment texts. As part of the experiments, Su et al. (2014) generate their data set by crawling thousands of Chinese comments related to clothing products. Firstly, they apply Word2Vec to cluster the synonyms that refer to the same product feature for the purpose of measuring its performance in the extraction of semantic features. Then, they use SVM to classify the comment texts. The best experimental results of the proposed Word2Vec- and SVM-based sentiment classification perform with 90% accuracy

with either of the methods used, whether it is the lexicon-based feature selection method or the part-of-speech-based method. According to Su et al. (2014) this performance exhibits the suitability of Word2Vec for sentiment analysis task.

In another study conducted by Hughes, Li, Kotoulas, and Suzumura (2017), convolutional neural networks (CNNs) and Word2Vec is used in order to automatically classify clinical text at a sentence level. The authors train the network on a dataset which provides a broad categorization of health information. Through a detailed evaluation, Hughes et al. (2017) demonstrate that their method performs 15% better than the various approaches commonly used in natural language processing tasks.

In the study of Ertugrul, Onal, and Acarturk (2017), the effect of regression on confidence scores in sentiment analysis using Turkish tweets is analyzed. They extract hand-crafted features including lexical features, emoticons and sentiment scores. The authors also use word embedding of tweets for regression and classification. Their findings show that using regression on confidence scores slightly improves sentiment classification accuracy. Moreover, combining word embeddings with handcrafted features reduces the feature size and performs better than alternative feature combinations (Ertugrul, Onal, & Acarturk, 2017).

In a study by Çoban (2017), audio and lyric features are used in the process of Turkish music genre classification. Textual features are extracted from lyrics through the use of a variety of feature extraction models such as Word2Vec and traditional Bag of Words. Experiments are conducted using the SVM classifier algorithm, which is followed by the analysis of the impact of feature selection and different feature groups on music genre classification. While considering lyrics-based music genre classification as a text classification task, the author also

examines the impact of term weighting method. According to the experimental results, textual features can be as effective as audio features in the Turkish music genre classification, especially when used with a supervised term weighting method. The study shows that the use of audio features alone results in 98% success rate, whereas using only lyrics features gives 94.32% accuracy score via the fourgram method. A combination of lyrics and audio features, on the other hand, reaches the highest success rate at 99.12%.

2.3 Paragraph vectors

Following Mikolov et al.'s (2013) approach for learning word embeddings—Word2Vec—Le and Mikolov (2014) propose another algorithm called Paragraph Vector. It is also known as Doc2vec because it comes in Gensim (Rehurek & Sojka, 2010) package with that name. This is an unsupervised method for learning distributed representations for documents and sentences. It is very similar to the Word2Vec method. Their difference is that there is a vector for each sentence or paragraph in the Doc2vec method. The Doc2vec method uses "Distributed Memory" model, Figure 1, in which it adds a memory vector to the standard language model that aims to capture the subject of the document.

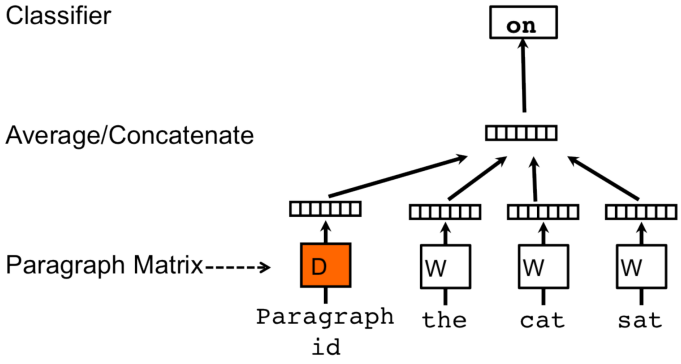

Figure 1. Distributed Memory model (adapted from Le and Mikolov (2014))

In their research, Le and Mikolov (2014) apply this method on a sentiment analysis over Stanford Sentiment Treebank and IMDB datasets. In their experiment, they demonstrate that Doc2vec can learn embedding of movie review texts which can be used for sentiment analysis. Accordingly, their method improves sentiment analysis results by 1.3% (or 15% relative improvement), over the best previous result.

Another study by Dai, Olah, and Le (2014) compare Doc2vec performance with other document modelling algorithms such as Latent Dirichlet Allocation. They benchmark the models on Wikipedia dataset and arXiv dataset. They state that the Doc2vec algorithm performs significantly better than other methods.

2.4 Music mood detection

Yang and Chen (2012) state that emotion-based music organization and retrieval is a logical way to access music data, for almost every piece of music expresses emotion. Early studies on music mood recognition use categorical labels such as happy or sad (Feng, Zhuang, & Pan, 2003). Feng and his colleagues employ an approach called Computational Media Aesthetics (CMA) to classify music emotion. Their approach

assumes that composers arise emotion by choreographing or managing expectation while artists translate musical intent into music language to create emotion. Thus, the music is analyzed in terms of how the music is made. In their approach, the music database is indexed on four musical moods; "happiness", "sadness", "anger" and "fear". And three properties—the relative tempo, the mean and the standard deviation of the average silence ratio (articulation)—are used to classify the mood using a back propagation neural network.

Another study looks into a million-scale music-listening dataset obtained from music-related Twitter hashtags (Hauger, Schedl, Kosir, & Tkalci, 2013). Hauger et al. (2013) present a large publicly available dataset which contains microblog-based music listening histories that include geographic, temporal, and other contextual information and give basic statistics about its composition. The authors perform a broad statistical study based on the correlation between music taste and day of the week, hour of day, and country in order to explain how their dataset helps detect new contextual music listening patterns (Hauger, Schedl, Kosir, & Tkalci, 2013). Hu and Downie (2010), on the other hand, presents a study that compares selected lyrics features and sound features to find effective features in classifying moods. The results of the study suggest that lyrics are the most effective features in classifying most of the moods.

There are studies comparable to mood detection such as genre identification, which has been used by Tzanetakis, Essl, and Cook (2002). Tzanetakis et al. (2002) apply Gaussian mixture models and diagonal covariance matrices, and they achieve 61% classification accuracy in ten genres. They use three features for classification: timbre texture, rhythmic content, and pitch content. Hamel and Eck (2010), on the other hand, propose a system that can automatically extract relevant features from

audio. Hamel and Eck (2010) use deep belief networks and a non-linear SVM classifier, which results in a classification score of 84.3% on the dataset of Tzanetakis et al. (2002). Using feedforward neural networks and k-nearest neighbor classifiers, McKay and Fujinaga (2004) classify the recordings by genre with features based on instrumentation, texture, rhythm, pitch statistics, dynamics, melody and chords. Consequently, they obtain a classification accuracy of 98% for root genres and 90% for leaf genre in a hierarchical taxonomy of 9 leaf genres (McKay & Fujinaga, 2004).

Barthet, Fazekas, and Sandler (2013) propose a study whose objective is to examine the available trends in music emotion recognition and offer insights that may help optimize music emotion recognition models. In their approach, they categorize emotions into a variety of classes, from happy to angry, relaxed to sad. Then an emotion classifier is created through selected machine learning techniques (Barthet, Fazekas, & Sandler, 2013). Afterwards, models generated through these techniques are employed to detect the emotion of a music piece provided as the input. To that effect, some machine learning algorithms have been employed to uncover the relationship between music features and mood labels including neural networks (Feng, Zhuang, & Pan, 2003), support vector machines (Laurier, Grivolla, & Herrera, 2008), fuzzy c-means classifier (Patra, Das, & Bandyopadhyay, 2013), and k-nearest neighbor (Dewi & Harjoko, 2010).

In the study by J. Liu, S. Liu, and Yang (2014), the role music plays in mood regulation is analyzed from a comprehensive perspective by employing the music emotion recognition models that have been trained from a Last.fm dataset comprised of 31,427 songs through the use of 190 music mood classes in total. Liu et al. (2014) propose how LJ2M (LiveJournal 2-million) can be used in understanding real-life

music listening behaviors. LJ2M consists of blog articles posted on the social blog site LiveJournal, along with tags that self-report a user's mood when posting and the musical piece that the user prefers for the post. As the underlying feature representation, Liu et al. (2014) employ the 12-D EchoNest timbre descriptor, and use SVM with the radial basis function kernel. Cross-validation results from the Last.fm dataset demonstrate that the accuracy obtained from 190 independent binary-classifiers in terms of AUC is 73.9%.

To the best of my knowledge, this is the first study that incorporates word and document embeddings for the classification of Turkish music according to their moods.

CHAPTER 3

METHODOLOGY

This section will introduce the methodology followed in the study. Below, an overview of the properties of Turkish language will be given, and then steps of acquiring the dataset, mood labeling of selected lyrics, and details of feature representation and weighting process will be explained. Additionally, an overview of the selected machine learning classifier algorithms, how these classifiers are utilized in this study, and details of the classification process will be presented.

3.1 Turkish language overview

In a study by Oflazer and Bozşahin (1994), Turkish is defined as an agglutinative language, in which word structures are formed by productive affixations of derivational and inflectional suffixes to stems. Using suffixes extensively causes morphological parsing of words to be complex, and leads to ambiguous lexical interpretations. It is possible to assign the meaning of a sentence in English to only one Turkish word. For instance, the meaning of English sentence 'I was not reading' can be expressed by only one Turkish word: 'read' is the stem and elements meaning 'not', '-ing', 'was', and 'I' are all suffixed to it respectively: 'Okumuyordum'. Turkish is an Altaic language that belongs to the Ural-Altaic family and derived from the Latin alphabet and consists of 21 consonants (b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y, z) and 8 vowels (a, e, ı, i, o, ö, u, ü); and seven of these letters are specific to the Turkish, modified from their original versions in the Latin alphabet (ç, ı, ş, ö, ü, ğ, İ).

3.2 Acquiring the data

This part describes the process of gathering the dataset of Turkish documents used for training of word embeddings, along with the Turkish lyrics dataset utilized in mood detection task. Firstly, more than 300 thousand multi-language lyrics are crawled over the Internet. Then, a language detection tool written in python language called langdetect (langdetect 1.0.7 : Python Package Index, 2016) detects the Turkish song lyrics among them, where the number of them reaches over 100 thousand. Along with the lyrics, a variety of Turkish documents are collected from Turkish Wikipedia and Turkish news websites to be used in the training of word embeddings. Consequently, more than 254 thousand Turkish Wikipedia documents and 2.3 million Turkish news documents are collected. At the end, without any further preprocessing like stemming or stop-words removal, this dataset consists of 570,842,387 tokens, and out of which 2,902,265 of them are unique tokens.

3.3 Mood tagging

Russell (1980) proposes the circumplex model of affect based on the two-dimensional model where the dimensions are “positive/negative valence” and “high/low arousal”. There are 28 affect words in Russell’s circumplex model which are shown in Figure 2.

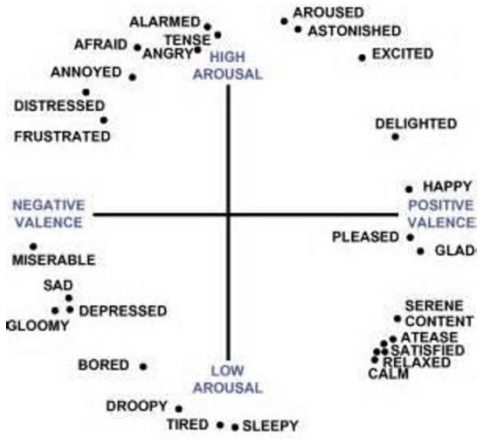

Figure 2. Circumplex model of affect (adapted from Russell (1980))

Several researchers have adopted a subset of Russell's taxonomy. For example, Hu and Downie (2010) use all the adjectives including calm, sad, glad, romantic, gleeful, gloomy, angry, mournful, dreamy, cheerful, brooding, aggressive, anxious, confident, hopeful earnest, cynical and exciting. Laurier, Grivolla, and Herrera (2008) and Song, Dixon, and Pearce (2012) use happy, sad, angry and relaxed as mood taxonomy. Patra, Das, and Bandyopadhyay (2013) adapt Russell's (1980) model into five clusters with three subclasses.

In consideration of the previous literature, in this study, four different mood categories are determined with three subclasses, which are shown in Table 1. Affect words that are similar to those of Russell's (1980) circumflex model are clustered.

Table 1. Four Emotion Cluster of Proposed Mood Taxonomy

	Cluster			
Mood	**1**	**2**	**3**	**4**
1	Happy	Calm	Sad	Angry
2	Delighted	Relaxed	Depressed	Annoyed
3	Excited	Satisfied	Gloomy	Tensed

So, the aim of this study is to automatically classify song lyrics into one of these four mood categories. Within this framework, I chose a set of popular artists and randomly selected more than 700 songs that belong to them for the manual annotation process. If a randomly-selected set of lyrics was a remix, acoustic or another adapted version of the original song, it was removed from the list in order to prevent redundancy of the data. Three people annotated the selected lyrics separately and at least two of the three annotators gave the same responses for the lyrics of 515 Turkish songs. Other lyrics were removed from the list. The mood class distribution of the 515 annotated lyrics is given in Table 2.

Table 2. Summary of Ground Truth Class Labels

Mood Category	**Number of Songs**
Happy	76
Calm	78
Sad	253
Angry	108

3.4 Pre-processing

Pre-processing of the datasets is one of the most important steps in text classification. Tokenization, elimination of stop-words and unnecessary characters, and stemming are the most commonly-utilized pre-processing methods.

In this study, I firstly conducted removal-based pre-processing. All non-alphabetic characters, punctuations, and non-printable characters were removed. Then, all characters were converted to lowercase and all dataset was tokenized.

Another widely-used pre-processing method is to remove stop-words, which are common frequent words (such as and, or, this) without any informative value for

text classification tasks. In this experiment, I created and used a Turkish stop-word list which consisted of 342 Turkish words.

For stemming, I employed a freely-available morphological analyzer for Turkish named TRMorph (Coltekin, 2010). TRmorph is a two-level Turkish morphological analyzer developed for the purpose of high availability and distributed with a license that allows anyone to use and modify it freely for different applications (Coltekin, 2010)

3.5 Feature representation and weighting

Before using machine learning classifiers, I transformed the lyrics dataset to lyrics vectors. First of all, I trained word embeddings with the dataset which consisted of more than 2.5 million Turkish documents. I employed two algorithms proposed for training word embeddings, namely Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) and GloVe (Pennington, Socher, & Manning, 2014).

Parameters for training the Word2Vec model were kept at their default values as suggested in the original papers. I trained the Word2Vec model through negative sampling with the parameter value 5 for negative, which specifies how many "noise words" should be drawn. Besides, I set 1 and 10 to the parameter of *min_count* in order to analyze its effect on the performance. This parameter is used to ignore all words with total frequency lower than the parameter's value in the entire corpus. In this experiment, size of the sliding window was set to 10—to the maximum distance between the current and predicted word within a sentence. I used 100, 200, 300 and 400 as the dimensionalities of the feature vectors of the words. Other parameters were kept at their default values. As a result, I obtained models of word embeddings trained using Word2Vec.

In the training process of GloVe word embeddings models, I used 100, 200, 300 and 400 as the dimensionalities of the feature vectors of the words as well. Rest of the parameters were kept at their default values, whereas the number of epochs was set to 20. Consequently, I obtained word embeddings trained using GloVe.

Following the process of training word embeddings, I applied three different approaches in order to generate lyrics vectors for the annotated lyrics. Firstly, I took the average of all words' word vectors that appeared in the song lyrics before filtering stop-words. In the second approach, I carried out the same process after filtering the stop-words. In the last approach, I calculated the average of the word vectors by multiplying them by their corresponding TF-IDF scores which were computed over 515 annotated song lyrics with three different threshold values—0.01, 0.001, and 0.00001 respectively. As a result, I obtained feature vectors for each set of song lyrics to be used in training of the selected machine learning classifiers, which were later used in automatic classification of songs into four mood categories.

For comparison, I also used Doc2Vec algorithm and bag-of-words approach based on TF-IDF scores in the Turkish music mood detection task.

3.6 Selected classifiers and classification

From the viewpoint of machine learning, the objective of text classification is to train classifiers over labelled documents and achieve classification on documents with unknown labels. There are many machine learning classifiers for text classification in the literature (Sebastiani F. , 2002). In consideration of high dimensionality, over-fitting characteristics and researches on text classification, this study was centered on four well-known text classification classifiers (SVM, Naïve Bayes, Random Forest,

and Logistic Regression). Detailed information about each selected classifier is presented in the following section.

3.6.1 Support Vector Machine (SVM)

SVM was introduced by Boser, Guyon, and Vapni (1992). This classification algorithm is extensively used in machine learning in the literature. It is comprised of both linear as well as non-linear algorithms and is based on statistical information theory and structural risk minimization. The linear SVM algorithm creates an infinite number of hyper-planes in order to separate data and selects a maximum margin hyper-plane among all these hyper-planes. If classes are not linearly separable, non-linear SVM is used to transfer data into a higher dimensional space. In this wise, the data becomes linearly separable (Sebastiani F. , 2005). Superior runtime behavior is the advantage of SVM during categorizing the new documents because dot product is calculated for each new document. On the other hand, it is a disadvantageous fact that since the similarity is generally calculated separately for each category, a document can be assigned to several categories.

In this study, SVC class in the scikit package (Pedregosa, et al., 2011) was utilized.

3.6.2 Naïve Bayes (NB)

Naïve Bayes (NB) is one of the most utilized classifiers in text classification. It is a simple probabilistic classifier based on Bayes theorem (Yildirim & Birant, 2014). NB assumes that the probability of a word's occurrence in a document is independent of the occurrence of other words in that document. The most prominent features of NB are high performance and easy implementation. The main

disadvantage is that dependencies between features cannot be modeled. The simple equation of the NB classifier is illustrated in the following (1):

$$P(c_j|d) = P(d|c_j)P(c_j)/P(d) \quad (1)$$

where $P(c_j|d)$ is the likelihood of sample d being in class of c_j, $P(d|c_j)$ is the likelihood of producing the sample d given the class of c_j, and the probability of occurrence of class of c_j and the probability of instance d occurring are $P(c_j)$ and $P(d)$, respectively.

In this study, GaussianNB class in the scikit package (Pedregosa, et al., 2011) was employed. GaussianNB is used when dealing with continuous data such as lyric vectors in this study. It assumes the likelihood of the features to be Gaussian.

3.6.3 Random Forest (RF)

It is known that the Random Forest (RF) algorithm is one of the most efficient classification methods. Proposed by Breiman (2001), RF is an ensemble learning method of decision trees, which is a combination of tree predictions, each tree being dependent on the values of an independently sampled random vector and with the same distribution for all the trees in the forest. Firstly, the subspaces of features are randomly selected at each node to grow branches of a decision trees (Xu, Guo, Y, & Cheng, 2012). Then, the training data is created and used to build each tree. Lastly, an RF classification model is created by combining individual trees. All input parameters are transmitted to each tree in the forest for the categorization of a document. Predictions for the classification label are collected from all the trees in the forest and the label with the highest rating is selected as the result (Kılınç, et al., 2017).

As stated in the study by Kılınç et al. (2017), RF is a fairly accurate classifier that works efficiently in big data sets with the capability of handling thousands of input properties without any deletion. Also, Kılınç et al (2017), define RF as a successful method for estimating missing data, and say that it maintains accuracy even when a large portion of the data is missing. RF also includes an experimental method to detect feature interactions and it may not work efficiently in a dataset that contains varying levels of categorical variables because random forests are prejudiced in favor of the attributes with more levels (Kılınç, et al., 2017).

In this study, RandomForestClassifier class in the scikit package (Pedregosa, et al., 2011) was employed with 100 trees.

3.6.4 Logistic Regression (LR)

Contrary to its name, Logistic Regression (LR) (a.k.a. logit, MaxEnt) is a linear model for classification rather than regression. In LR, the probabilities which explain the possible outcomes of a single trial are modeled using a logistic function (Murphy, 2012).

In this study, LogisticRegression class in the scikit package (Pedregosa, et al., 2011) was used.

CHAPTER 4

RESULTS

In this study, micro and macro-averaged f1 score are used together with 10-fold cross validation for performance measurements. F1 score (a.k.a. F-score, F-measure) is a measure of the accuracy of a test, which takes values in between 0 and 1 where 1 is the best value for an F1 score, and 0 is the worst. It considers both the precision and the recall scores obtained from the runs on the test data. It is calculated by using equation (2):

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2}$$

Precision is the ratio of all positive predictions that are correct, whereas recall is the ratio of all positive observations that are correctly predicted. Precision and Recall are calculated using equations (3) and (4), respectively:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{3}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{4}$$

Macro-averaging is used to calculate metrics for each label in order to obtain their unweighted mean. However, macro-averaging does not take label imbalance into account. On the other hand, micro-averaging counts the total true positives (TP), false negatives (FN) and false positives (FP), and thus calculates metrics globally. Due to the fact that micro-averaging takes label imbalance into consideration, micro-averaged f1 score is a more appropriate measure for this study.

4.1 Performance of proposed approach

At first, the proposed approach was applied on Turkish music mood detection by using word vectors created via Word2Vec and GloVe word embeddings algorithms separately. Then, the same process was conducted with the methods of Doc2Vec algorithm and bag-of-words approach based on TF-IDF scores. Three different n-gram range parameters—unigram, bigram, and trigram—were used for the bag-of-words method. In the proposed approach and the Doc2Vec method, the vector size is used as 100. The results obtained using different parameters are given in the Table 3.

Table 3. Performance Comparisons

	Classifier							
	SVM		NB		RF		LR	
	F1 Score							
Method	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**
Proposed Approach using Word2Vec Averages	46.95%	42.55%	46.71%	39.97%	**52.18%**	26.92%	46.31%	39.72%
Proposed Approach using Word2Vec Averages + Stop-words Removed	49.12%	42.55%	49.03%	39.97%	51.97%	26.92%	46.54%	39.72%
Proposed Approach using Word2Vec Averages + TF-IDF (Unigram, Threshold: 0.00001)	48.96%	16.43%	45.10%	39.73%	52.02%	25.09%	43.46%	38.19%
Proposed Approach using Word2Vec Averages + TF-IDF (Unigram, Threshold: 0.001)	48.96%	16.43%	45.10%	39.73%	52.02%	25.09%	43.46%	38.19%
Proposed Approach using Word2Vec Averages + TF-IDF (Unigram, Threshold: 0.01)	48.96%	16.43%	44.71%	39.33%	50.86%	23.07%	43.46%	38.19%
Proposed Approach using GloVe Averages	40.18%	36.17%	40.65%	33.32%	51.59%	23.83%	49.28%	32.91%
Proposed Approach using GloVe Averages + Stop-words Removed	41.72%	36.71%	39.20%	33.60%	50.23%	23.78%	49.85%	35.95%
Proposed Approach using GloVe Averages + TF-IDF (Unigram, Threshold: 0.00001)	45.41%	41.15%	38.43%	34.41%	50.84%	25.37%	46.21%	40.58%
Proposed Approach using GloVe Averages + TF-IDF (Unigram, Threshold: 0.001)	45.41%	41.15%	38.43%	34.41%	50.84%	25.37%	46.21%	40.58%
Proposed Approach using GloVe Averages + TF-IDF (Unigram, Threshold: 0.01)	45.41%	41.15%	38.43%	34.41%	50.62%	25.37%	46.21%	40.58%
Doc2Vec	50.55%	34.96%	39.02%	32.45%	50.07%	19.85%	39.61%	34.26%
Bag-of-Words Based On TF-IDF (unigram)	48.93%	27.62%	47.18%	25.91%	49.52%	18.38%	49.74%	20.33%
Bag-of-Words Based On TF-IDF (bigram)	49.73%	19.70%	47.78%	29.39%	49.14%	17.54%	48.95%	16.96%
Bag-of-Words Based On TF-IDF (trigram)	48.95%	16.96%	47.60%	28.94%	49.14%	17.90%	48.75%	16.38%

As seen in the Table 3, Word2Vec performs better than GloVe for the dataset used in this study. The best score achieved is 52.18%, which was obtained using the RF classifier through the proposed approach with the lyrics vectors computed by averaging the word vectors generated via Word2Vec algorithm. Removal of the stop-words has some negative effect on the score obtained from the RF classifier, but it is evident that there is an improvement in the scores of other classifiers. In addition, incorporation of TF-IDF scores into the lyrics vector computation process seems to reduce the score slightly.

The best score obtained from the approaches used in this study is 1.63% higher (or 3.22% relatively) than the best score obtained from the Doc2Vec and 2.44% higher (or 4.90% relatively) than the best score obtained from the bag-of-words approach based on TF-IDF scores. This result shows that the proposed method is effective and inclusion of word embeddings in the Turkish text classification process improves the performance.

4.2 Evaluation of dimensionality and minimum word count

In this step, the consequential effects of using different dimensions and word counts for word embeddings are investigated. The following Table 4 demonstrates the changes in the performance of the best-performed approach (Proposed Approach using Word2Vec) with respect to dimensions and word counts:

Table 4. Evaluation of Stemming on Proposed Approach Using Word2Vec vs. Doc2vec Method

		Classifier							
		SVM		**NB**		**RF**		**LR**	
		F1 Score							
Dimension	**Min-word**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**
100	10	46.95%	42.55%	46.71%	39.97%	**52.18%**	26.92%	46.31%	39.72%
200	10	46.55%	40.87%	46.18%	39.00%	50.64%	22.52%	48.29%	39.66%
300	10	47.90%	42.93%	46.36%	39.97%	50.62%	21.97%	46.39%	38.74%
400	10	47.31%	43.00%	44.43%	38.25%	50.86%	21.52%	48.06%	39.56%
100	1	47.37%	41.87%	46.86%	38.94%	51.61%	24.56%	46.95%	40.05%
200	1	46.53%	40.92%	44.75%	38.63%	51.21%	23.24%	48.68%	40.99%
300	1	45.15%	40.55%	44.38%	37.46%	50.80%	22.76%	46.32%	39.53%
400	1	46.90%	42.44%	44.54%	38.78%	51.20%	23.40%	43.59%	36.01%

The Table 4 shows that the increase in dimension size of word embeddings causes very little negative effect on the result. And, the decrease in the minimum word count has no significant effect on the result. Consequently, the best result is obtained with 100-dimension size and 10 minimum word count.

4.3 Evaluation of stemming

Since Turkish is an agglutinative language, in which word structures are formed by productive affixations of derivational and inflectional suffixes to word roots, I investigated the effects of stemming on the classification performance. Because the Doc2Vec method has the closest score to the proposed method, the effects of stemming on the result of Doc2Vec method were also examined. The following Table 5 shows the comparison of performance results obtained by using stemmed and non-stemmed lyrics in Doc2Vec method and the proposed approach:

Table 5. Evaluation of Word Embeddings Dimensionality and Minimum Word Count on Proposed Approach Using Word2Vec

		Classifier							
		SVM		**NB**		**RF**		**LR**	
		F1 Score							
Method	**Stemming**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**
Proposed Approach Using Word2Vec Averages	TRUE	48.18%	44.37%	46.37%	39.29%	**54.36%**	28.41%	46.43%	41.22%
Proposed Approach Using Word2Vec Averages	FALSE	46.95%	42.55%	46.71%	39.97%	52.18%	26.92%	46.31%	39.72%
Doc2Vec	TRUE	48.85%	33.64%	48.47%	36.56%	49.88%	19.21%	47.64%	40.58%
Doc2Vec	FALSE	50.55%	34.96%	39.02%	32.45%	50.07%	19.85%	39.61%	34.26%

The Table 5 shows that stemming improves the performance of the proposed method. However, stemming does not have a significant effect on the results obtained by Doc2Vec method.

As a result, the proposed method using lyrics vectors calculated via averaging the respective word vectors generated via Word2Vec method achieves a score 3.81% higher (or 7.54% relatively) than the best score obtained from the Doc2Vec method and 4.62% higher (or 9.29% relatively) than the best score obtained from bag-of-words approach based on TF-IDF scores.

CHAPTER 5

CONCLUSION

In this research, an approach that incorporates word embeddings into Turkish text classification process was investigated. To that end, various parameters and settings in generating word embeddings were assessed in Turkish music mood detection task. In order to train the word embeddings, two very popular word embeddings algorithms were employed in our approach; namely Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) and GloVe (Pennington, Socher, & Manning, 2014). Then, lyric/paragraph vectors of the labeled lyrics were created by using these word embeddings. For comparison, Turkish music mood detection was also performed through the Doc2Vec algorithm and popular bag-of-words (up-to including Trigram features) approach which uses TF-IDF scores. Finally, Turkish lyrics mood detection was conducted by applying the selected machine learning classifier algorithms that use lyrics vectors as features, and the results were compared for accuracy. Micro and macro F1 scores were used as the performance measures due to the imbalanced class distribution experienced in the labeled lyrics dataset. The consequential effects of stemming of words into their roots, and filtering of the stop-words were also investigated. The results of the study show that the score of the proposed approach is 3.81%, and 4.62% higher (7.54%, and 9.29% improvement) than the best score obtained from Doc2Vec and bag-of-words methods, respectively.

These results support the fact that word embeddings are effective and efficient representations of the words that may be utilized for text classification

purposes, which is consistent with the findings of the previous studies. Besides, training word vectors with a large collection of data, compared to even more powerful approaches like training and utilizing paragraph vectors, can obtain better results. So, incorporating word embeddings trained with large amounts of textual data into the Turkish text classification process improves its performance.

Music companies and individuals can benefit from the methodology proposed in the study. For example, managing rapidly-growing collections of digital music and building more reliable music recommendation systems have always been a challenge for companies which provide music services, such as Spotify and Apple Music. Also, it is time consuming and difficult for listeners, shop owners, advertisers, and filmmakers to manually select songs that satisfy a particular mood requirement. Using the proposed method can help these companies and individuals to find the kind of music they are looking for and better manage their music collections.

CHAPTER 6

LIMITATIONS AND FUTURE WORK

Mood perceptions of songs may vary from one listener to another, depending on their emotional state. Whereas a song can be classified as sad by a listener, another listener may classify the song as aggressive. This situation may create an obstacle to this research and further researches on related topics. To eliminate this problem in this research, the lyrics were classified by at least three different people separately. On the other hand, it should be noted that reliability of such a research can be increased with the number of annotators. The annotators consisted of young people whose ages ranged from 20 to 35. Although the group of annotators in this research could not represent the whole community, resulting classes can be considered consistent within themselves. Besides, the success of machine learning applications often increases when the size of the dataset expands. In this study, the annotated dataset used consisted of 515 lyrics. This figure can be considered relatively small. Therefore, it is recommended that the size of labeled dataset be kept as large as possible in future studies. Moreover, utilizing crowdsourcing for emotion classification with more annotators would result in achieving a more reliable classification of the songs.

In addition, because a piece of music consists of lyrics and sounds, including lyric features and audio features of songs together in further researches should give better and more reliable results.

APPENDIX A

PRE-PROCESSING FUNCTION CODES

```
def strip_tr(str):
    # türkçe harfleri küçük harfe çeviriyoruz
    HARFDIZI = [(u'İ', u'i'), (u'Ğ', u'ğ'), (u'Ü', u'ü'), (u'Ş', u'ş'),
(u'Ö', u'ö'), (u'Ç', u'ç'), (u'I', u'ı')]
    for aranan, harf in HARFDIZI:
        str = str.replace(aranan, harf)

    # tüm harfleri küçük harfe çeviriyoruz
    str = str.lower()

    # noktalama işaretlerini boşluk karakteriyle değiştiriyoruz
    exclude = set(string.punctuation)
    for x in exclude:
        str = str.replace(x, ' ')
    # str = ''.join(ch for ch in str if ch not in exclude)

    # rakamları boşluk karakteriyle değiştiriyoruz
    exclude = set('0123456789')
    for x in exclude:
        str = str.replace(x, ' ')
    # str = ''.join(ch for ch in str if ch not in exclude)

    # başka alfabeye ait olan harf barındıran kelimeleri ve kelime
aralarındaki mükerrer boşlukları siliyoruz
    regex_w = re.compile(ur'[^a-zığüşöç ]')
    tmp = []
    for word in str.split():
        if regex_w.search(word) is None and len(word) > 1:
            tmp.append(word)
    str = " ".join(tmp)

    return str
```

APPENDIX B

TURKISH STEMMER CLASS CODES

```
# encoding=utf8
import foma
import re
import string

class TRStemmer:
    def __init__(self, cache=False):
        self.stemmer = foma.read_binary('stem2.fst')
        self.s = re.compile("<[^>]*>")
        self.t = re.compile("<([^>]*)>")
        self.cache = {}
        self.c = cache

    def get_stem(self, word):
        stem = None
        for result in self.stemmer.apply_up(word):
            tmp_s = self.s.sub("", result)
            tmp_t = self.t.findall(result)

            if len(tmp_t[0].split(":")) > 1:
                if tmp_t[0].split(":")[-1] == "mredup":
                    continue
                if tmp_t[0].split(":")[0] == "Num":
                    continue

            if tmp_t[0].startswith(u"Prn:pers"):
                return (tmp_s, tmp_t[0])
            if stem is None:
                stem = (tmp_s, tmp_t[0])
        return stem

    def stemmize(self, sentence):
        stemmed = u""
        for word in re.sub(' +', ' ', sentence).split(" "):
            if self.c and self.cache.has_key(word):
                if self.cache[word] is not None:
                    stemmed += self.cache[word] + u" "
            else:
                stem = self.get_stem(word)
                if stem:
                    # if stem[1] == "Adv":
                    #    #eğer stem zarf ise geçiyoruz
                    #    continue
                    # elif stem[1] == "Punc":
                    if stem[1] == "Punc":
                        # eğer noktalama işareti ise geçiyoruz
                        self.cache[word] = None
                        continue
                    elif stem[1] == "Num:ara":
                        # eğer numara ise geçiyoruz
                        self.cache[word] = None
                        continue
                    else:
                        self.cache[word] = stem[0]
                        stemmed += stem[0] + " "
```

```
            else:
                self.cache[word] = word
                stemmed += word + u" "

        return stemmed.strip()
```

APPENDIX C

LIST OF USED STOP-WORDS

a, acaba, altı, altmış, ama, ancak, arada, artık, asla, aslında, ayrıca, az, b, bana, bazen, bazı, bazıları, bazısı, belki, ben, benden, beni, benim, beri, beş, bile, bilhassa, bin, bir, biraz, birçoğu, birçok, birçokları, biri, birisi, birkaç, birkaçı, birkez, birşey, birşeyi, biz, bizden, bize, bizi, bizim, böyle, böylece, bu, buna, bunda, bundan, bunlar, bunları, bunların, bunu, bunun, burada, bütün, c, ç, çoğu, çoğuna, çoğunu, çok, çünkü, d, da, daha, dahi, dan, de, den, defa, değil, demek, diğer, diğeri, diğerleri, diye, doksan, dokuz, dolayı, dolayısıyla, dört, e, edecek, eden, ederek, edilecek, ediliyor, edilmesi, ediyor, eğer, elbette, elli, en, etmesi, etti, ettiği, ettiğini, f, fakat, falan, felan, filan, g, gene, gereği, gerek, gibi, göre, ğ, h, hala, hâlâ, halde, halen, hangi, hangisi, hani, hatta, hem, henüz, hep, hepsi, hepsine, hepsini, her, herbiri, herhangi, herkes, herkese, herkesi, herkesin, hey, hiç, hiçkimse, hiçbir, hiçbiri, hiçbirine, hiçbirini, hişt, ı, i, için, içinde, iki, ile, ilgili, ise, işte, itibaren, itibariyle, j, k, kaç, kadar, karşın, katrilyon, kendi, kendilerine, kendine, kendini, kendisi, kendisine, kendisini, kez, kırk, ki, kim, kimden, kime, kimi, kimin, kimisi, kimse, l, m, madem, mı, mi, miyim, misin, misiniz, milyar, milyon, mu, muyum, musunuz, mü, müyüm, müsünüz, n, nasıl, ne, nekadar, nezaman, neden, nedenle, nedir, nerde, nerede, nereden, nereye, nesi, neyse, nın, niçin, nin, niye, nun, nün, o, olan, olarak, oldu, olduğu, olduğunu, olduklarını, olmadı, olmadığı, olmak, olması, olmayan, olmaz, olsa, olsun, olup, olur, olursa, oluyor, on, ona, ondan, onlar, onlara, onlardan, onları, onların, onu, onun, orada, otuz, oysa, oysaki, ö, öbür, öbürü, ön, önce, öte, ötürü, öyle, p, pek, r, rağmen, s, sadece, sana, sanki, sekiz, seksen, sen, senden, seni,

senin, siz, sizden, size, sizi, sizin, son, sonra, ş, şayet, şekilde, şey, şeyden, şeye, şeyi, şeyler, şimdi, şöyle, şu, şuna, şunda, şundan, şunlar, şunları, şunu, şunun, t, ta, taa, tabi, tabii, tam, tamam, tamamen, tarafından, trilyon, tüm, tümü, u, un, ü, üç, ün, üzere, v, var, vardı, ve, veya, veyahut, y, ya, yada, yani, yapacak, yapılan, yapılması, yapıyor, yapmak, yaptı, yaptığı, yaptığını, yaptıkları, ye, yedi, yerine, yetmiş, yı, yi, yine, yirmi, yoksa, yu, yüz, z, zaten, zira

REFERENCES

Akkus, B. K., & Cakici, R. (2013, August). Categorization of Turkish news documents with morphological analysis. In Dey A., Krause, S., Nikolova, I., Vecchi, E., Bethard, S., Nakov, P. I., & Xu, F. (Eds.), *Proceedings of the Student Research Workshop* (pp. 1-6). Sofia, Bulgaria: Association for Computational Linguistics.

Alparslan, E., Karahoca, A., & Bahsi, H. (2011). Classification of confidential documents by using adaptive neurofuzzy inference systems. *Procedia Computer Science*, *3*, 1412-1417. doi: 10.1016/j.procs.2011.01.023

Al-Radaideh, Q. A., Al-Eroud, A. F., & Al-Shawakfa, E. M. (2012). A hybrid approach to detecting alerts in Arabic e-mail messages. *Journal of Information Science*, *38*(1), 87-99.

Amancio, D. R., Comin, C. H., Casanova, D., Travieso, G., Bruno, O. M., Rodrigues, F. A., & da Fontoura Costa, L. (2014). A systematic comparison of supervised classifiers. *PloS one*, *9*(4), 94-137.

Amasyali, M. F., & Beken, A. (2009, April). Measurement of Turkish word semantic similarity and text categorization application. In Zelnio, E.G. & Garber, F.D. (Eds.), *Proceedings of IEEE 17th Signal Processing and Communications Applications Conference* (pp. 1-4). Antalya, Turkey: IEEE.

Amasyali, M. F., & Diri, B. (2006). Automatic Turkish text categorization in terms of author, genre and gender. In Kop, C., Fliedl, G., Mayr, H. C., & Métais, E. (Eds.), *Proceedings of 11th International Conference on Applications of Natural Language to Information Systems* (pp. 221-226). Klagenfurt, Austria: Springer

Barthet, M., Fazekas, G., & Sandler, M. (2012, June). Music emotion recognition: From content-to context-based models. In Aramaki, M., Barthet, B., Kronland-Martinet, R., & Ystad, S. (Eds.), *Proceedings of International Symposium on Computer Music Modeling and Retrieval* (pp. 228-252). Berlin, Heidelberg: Springer.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In Haussler, D. (Ed.), *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). Pittsburgh, PA, USA: ACM Press.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. *Ann Arbor MI*, *48113*(2), 161-175.

Chen, S. Y., & Liu, X. (2004). The contribution of data mining to information science. *Journal of Information Science*, *30*(6), 550-558.

Ciravegna, F., Gilardoni, L., Lavelli, A., Mazza, S., Black, W. J., Ferraro, M., ... & Rinaldi, F. (2000, August). Flexible text classification for financial applications: the FACILE system. In Horn, W. (Ed.), *Proceedings of the 14th European Conference on Artificial Intelligence* (pp. 696-700). Amsterdam, Netherlands: IOS Press.

Coltekin, C. (2010, May). A freely available morphological analyzer for Turkish. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., ... Tapias, D. (Eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* (Vol. 2, pp. 19-28). Valletta, Malta: European Language Resources Association (ELRA)

Çataltepe, Z., Turan, Y., & Kesgin, F. (2007, June). Turkish document classification using shorter roots. Paper presented at the 15th Signal Processing and Communications Applications Conference (SIU). Eskişehir, Turkey. doi: 10.1109/SIU.2007.4298757

Çoban, Ö. (2017). Turkish music genre classification using audio and lyrics features. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *0*. doi: 10.19113/sdufbed.88303.

Dai, A. M., Olah, C., & Le, Q. V. (2014, December). Document embedding with paragraph vectors. Paper presented at Proc. of NIPS 2014 in Deep Learning and Representation. Montreal, Canada. Retrieved from https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44894.pdf

Danilak, M. M. (2016). langdetect (Version 1.0.7) [Computer software]. Retrieved January 01, 2017, from https://github.com/Mimino666/langdetect

Dewi, K. C., & Harjoko, A. (2010, August). Kid's song classification based on mood parameters using k-nearest neighbor classification method and self organizing map. Paper presented at the 2010 International Distributed Framework and Applications Conference (DFmA). Jogjakarta, Indonesia.

El Kourdi, M., Bensaid, A., & Rachidi, T. E. (2004, August). Automatic Arabic document categorization based on the naïve bayes algorithm. In Farghaly, A., & Oroumchian, F. (Eds.), *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages* (pp. 51-58). Geneva, Switzerland: Association for Computational Linguistics.

Ertugrul, A. M., Onal, I., & Acarturk, C. (2017, June). Does the strength of sentiment matter? A regression based approach on Turkish social media. In Frasincar, F., Ittoo, A., Nguyen, L. M., & Métais, E. (Eds.), *Proceedings of International Conference on Applications of Natural Language to Information Systems* (pp. 149-155). Liège, Belgium: Springer.

Feng, Y., Zhuang, Y., & Pan, Y. (2003, July). Popular music retrieval by detecting mood. In Callan, J., Crestani, F., & Sanderson, M. (Eds.), *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 375-376). Toronto, Canada: ACM.

Gunal, S. (2012). Hybrid feature selection for text classification. *Turkish Journal of Electrical Engineering & Computer Sciences*, *20*(Sup. 2), 1296-1311. doi:10.3906/elk-1101-1064

Güran, A., Akyokuş, S., Bayazıt, N. G., & Gürbüz, M. Z. (2009, June). Turkish text categorization using n-gram words. In Yildirim, T., Altas, H. A., Okumus, H. I., & Ozkop, E. (Eds.), *Proceedings of INISTA 2009 International Symposium on INnovations in Intelligent SysTems and Applications* (pp. 369-373). Trabzon, Turkey: Karadeniz Technical University Press.

Hamel, P., & Eck, D. (2010, August). Learning features from music audio with deep belief networks. In Downie, J. S. & Veltkamp, R. C. (Eds.), *Proceedings of the 11th International Society for Music Information Retrieval Conference* (pp. 339-344). Utrecht, Netherlands: International Society for Music Information Retrieval.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146-162.

Hauger, D., Schedl, M., Košir, A., & Tkalcic, M. (2013, November). The million musical tweets dataset: What can we learn from microblogs. In Britto, A. D. S., Jr., Gouyon, F., & Dixon, S. (Eds.), *Proceedings the 14th International Society for Music Information Retrieval Conference* (pp. 189-194). Curitiba, Brazil: International Society for Music Information Retrieval.

Hu, X., & Downie, J. S. (2010, August). When lyrics outperform audio for music mood classification: A feature analysis. In Hirata, K., Tzanetakis, G., & Yoshii, K. (Eds.), *Proceedings of the 10th International Society for Music Information Retrieval Conference* (pp. 619-624). Kobe, Japan: International Society for Music Information Retrieval.

Hughes, M., Li, I., Kotoulas, S., & Suzumura, T. (2017). Medical text classification using convolutional neural networks. In Randell, R., Cornet, R., McCowan, C., Peek, N., & Scott, P. (Eds.), *Informatics for Health: Connected Citizen-Led Wellness and Population Health* (pp. 246-250). Amsterdam, Netherlands: IOS Press.

Kılınç, D., Özçift, A., Bozyigit, F., Yıldırım, P., Yücalar, F., & Borandag, E. (2017). TTC-3600: A new benchmark dataset for Turkish text categorization. *Journal of Information Science*, *43*(2), 174-185.

Laurier, C., Grivolla, J., & Herrera, P. (2008, December). Multimodal music mood classification using audio and lyrics. In Wani, M. A., Chen, X., Casasent, D., Kurgan, L. A., Hu, T., & Hafeez, K. (Eds.), *Proceedings of 2008 Seventh International Conference on Machine Learning and Applications* (pp. 688-693). San Diego, CA, USA: IEEE. doi: 10.1109/ICMLA.2008.96.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In Phung, D. & Li, H. (Eds.), *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (pp. 1188-1196). Beijing, China: PMLR

Liu, J. Y., Liu, S. Y., & Yang, Y. H. (2014, July). LJ2M dataset: Toward better understanding of music listening behavior and user mood. Paper presented at the International Conference on Multimedia and Expo 2014 (ICME). Chengdu, China. doi: 10.1109/ICME.2014.6890172

McCallum, A., Rosenfeld, R., Mitchell, T. M., & Ng, A. Y. (1998, July). Improving text classification by shrinkage in a hierarchy of classes. In Shavlik, J. W. (Ed.), *Proceedings of ICML* (Vol. 98, pp. 359-367). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

McKay, C., & Fujinaga, I. (2004, October). Automatic genre classification using large high-level musical feature sets. Paper presented at ISMIR 2004. Barcelona, Spain. Retrieved from http://jmir.sourceforge.net/publications/ISMIR_2004_Bodhidharma.pdf

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning, neural and statistical classification. Ellis Horwood Limited.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, June). Efficient estimation of word representations in vector space. Paper presented at the International Conference on Learning Representations: Workshops Track. Scottsdale, Arizona. Retrieved from https://arxiv.org/pdf/1301.3781.pdf

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K.Q. (Eds.), *Proceedings of Advances in neural information processing systems* (pp. 3111-3119). Lake Tahoe, Nevada: Curran Associates, Inc.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Oflazer, K. & Bozşahin, H. C. (1994, June). Turkish natural language processing initiative: An overview. Paper presented at the Third Turkish Symposium on Artificial Intelligence. Ankara, Turkey. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.8063

Özgür, L., Güngör, T., & Gürgen, F. (2004). Adaptive anti-spam filtering for agglutinative languages: A special case for Turkish. *Pattern Recognition Letters*, *25*(16), 1819-1831.

Patra, B. G., Das, D., & Bandyopadhyay, S. (2013). Unsupervised approach to Hindi music mood classification. In Prasath, R. & Kathirvalavakumar, T. (Eds.) *Proceedings of Mining Intelligence and Knowledge Exploration* (pp. 62-69). Tamil Nadu, India: Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825-2830.

Peng, F., Huang, X., Schuurmans, D., & Wang, S. (2003, July). Text classification in Asian languages without word segmentation. In Adachi, J. (Ed.), *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11* (pp. 41-48). Sapporo, Japan: Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Wu, D., Carpuat, M., Carreras, X., & Vecchi, E. M. (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Vol. 14, pp. 1532-1543). Doha, Qatar: Association for Computational Linguistics.

Read, J. (2005, June). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Callison-Burch, C. & Wan, S., (Eds.), *Proceedings of the ACL student research workshop* (pp. 43-48). Michigan, USA: Association for Computational Linguistics.

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In Witte, R., Cunningham, H., Patrick, J., Beisswanger, E., Buyko, E., Hahn, U., Verspoor, K., & Coden, A. R. (Eds.), *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45-50). Valletta, Malta: ELRA.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology, 39*(6), 1161-1178.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

Sebastiani, F. (2005). Text categorization. In Zanasi, A., (Ed.), *Proceedings of Text mining and its applications to intelligence, Crm and knowledge management* (pp. 109–129). Southampton, UK: WIT Press

Shaalan, K., & Oudah, M. (2014). A hybrid approach to Arabic named entity recognition. *Journal of Information Science*, *40*(1), 67-87.

Song, Y., Dixon, S., & Pearce, M. (2012, October). Evaluation of musical features for emotion classification. In Gouyon, F., Herrera, P., Martins, L. G., & Müller, M. (Eds.), *Proceedings of the 13th International Society for Music Information Retrieval Conference* (pp. 523-528). Porto, Portugal: International Society for Music Information Retrieval.

Su, Z., Xu, H., Zhang, D., & Xu, Y. (2014, September). Chinese sentiment classification using a neural network tool—Word2vec. Paper presented at the 2014 International Multisensor Fusion and Information Integration for Intelligent Systems Conference. Beijing, China. doi: 10.1109/MFI.2014.6997687

Sun, A., & Lim, E. P. (2001, November). Hierarchical text classification and evaluation. Paper presented at the International Conference on Data Mining. San Jose, CA, USA. doi: 10.1109/ICDM.2001.989560

Torunoğlu, D., Çakirman, E., Ganiz, M. C., Akyokuş, S., & Gürbüz, M. Z. (2011, June). Analysis of preprocessing methods on classification of Turkish texts. Paper presented at the 2011 International Symposium on Innovations in Intelligent Systems and Applications. Istanbul, Turkey. doi: 10.1109/INISTA.2011.5946084

Turner, V., Gantz, J. F., Reinsel, D., & Minton, S. (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*. Retrieved January 05, 2017, from https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, *10*(5), 293-302.

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, *50*(1), 104-112.

Xu, B., Guo, X., Ye, Y., & Cheng, J. (2012). An improved random forest classifier for text categorization. *JCP*, *7*(12), 2913-2920.

Yang, J., Qu, Z., & Liu, Z. (2014). Improved feature-selection method considering the imbalance problem in text categorization. *The Scientific World Journal*, *2014*, 1-17. doi:10.1155/2014/625342

Yang, Y. H., & Chen, H. H. (2012). Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *3*(3), 1-30.

Yildirim, P., & Birant, D. (2014, June). Naive bayes classifier for continuous variables using novel method (NBC4D) and distributions. Paper presented at the 2014 International Symposium on Innovations in Intelligent Systems and Applications. Alberobello, Italy. doi: 10.1109/INISTA.2014.6873605