MULTIWORD EXPRESSION DETECTION USING WORD VECTOR REPRESENTATIONS

TANSU TAŞÇIOĞLU

SEPTEMBER 2019

MULTIWORD EXPRESSION DETECTION USING WORD VECTOR REPRESENTATIONS

A THESIS SUBMITTED TO

IZMIR UNIVERSITY OF ECONOMICS

GRADUATE SCHOOL

BY

TANSU TAŞÇIOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

IN IZMIR UNIVERSITY OF ECONOMICS GRADUATE SCHOOL

SEPTEMBER 2019

Approval of Izmir University of Economics Graduate School

Prof. Dr. Mehmet Efe Biresselioğlu
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Kaya Oğuz
Head of Department

We have read the thesis entitled **"MULTIWORD EXPRESSION DETECTION USING WORD VECTOR REPRESENTATIONS"** completed by **TANSU TAŞÇIOĞLU** under supervision of **Assoc. Prof. Dr. Senem Kumova Metin** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Senem Kumova Metin
Supervisor

Examining Committee Members

Date: 25.09.2019

Assoc. Prof. Dr. Senem Kumova Metin
Dept. of Software Engineering, İUE

Asst. Prof. Dr. Kaya Oğuz
Dept. of Computer Enginnering, İUE

Assoc. Prof. Dr. Tarık Kışla
Dept. of Computer Education and
Instructional Technologies, Ege U.

ABSTRACT

MULTIWORD EXPRESSION DETECTION USING WORD VECTOR REPRESENTATIONS

TANSU TAŞÇIOĞLU
M.S. in Computer Engineering
Izmir University of Economics Graduate School
Supervisor: Assoc. Prof. Dr. Senem Kumova Metin
September 2019

Multiword expressions (MWE) are statements in which two or more words are combined traditionally in language. In most of multiword expressions, words combine losing/changing their own meanings in order to create a new one. In recent natural language processing studies, the meanings/senses of the words/word combinations are expressed by word vector representations (word embeddings). In vector representation, it is assumed that the neighbouring words hold the information regarding to the given target word in language.

The aim of this thesis is to explore the use of word representations in multiword expression detection in Turkish. We assumed that as the words combine to build up an MWE, they modify or lose their meanings resulting with a change in the vector representation.

In this thesis, word vectors of MWE candidates (both stemmed and surface forms) and composing words are built up by five different representation methods. The vector representation of MWE candidates are given as inputs to ten different types of classifiers. The classification performance is measured by F1 score with 5-fold cross validation. The experimental results showed that stemming does not improve the performance of MWE extraction when vector representations are used. In addition, it is observed that there exists no classification method that outperforms the others continuously in MWE detection experiments.

Keywords: Multiword expression, Word Vector Representation, Turkish.

ÖZ

SÖZCÜK TEMSİLLERİ KULLANARAK ÇOK SÖZCÜKLÜ İFADE TESPİTİ

TANSU TAŞÇIOĞLU
Bilgisayar Mühendisliği, Yüksek Lisans
Lisansüstü Eğitim Enstitüsü
Tez Danışmanı: Doç. Dr. Senem Kumova Metin
Eylül 2019

Çok sözcüklü ifadeler iki ve ya daha fazla sözcüğün geleneksel olarak dilde bir araya geldiği ifadelerdir. Çok sözcüklü ifadelerin çoğunda, kelimeler yeni bir anlam oluşturmak için bir araya gelirken kendi anlamlarını kaybederler. Son yapılan doğal dil işleme çalışmalarında, kelimelerin/kelime kombinasyonlarının anlamı sözcük temsilleri ile ifade edilir. Bu yaklaşımda, komşu sözcüklerin verilen hedef kelime ile ilgili bilgiyi taşıdığı kabul edilir.

Bu tez çalışmasının amacı, Türkçe'de çok sözcüklü ifadelerin tespitinde sözcük temsillerinin kullanımını araştırmaktır. Kelimeler çok sözcüklü ifadeler oluşturmak için bir araya geldiğinde vektör temsillerinde anlam değişikliği ya da kaybı olduğu kabul edilir.

Bu tezde, çok sözcüklü ifade adaylarının ve adayları oluşturan sözcüklerin sözcük temsil vektörleri (gövde ve yüzeysel form) beş farklı temsil yöntemi ile oluşturulmuştur. Çok sözcüklü ifade adaylarının vektör temsili on farklı sınıflandırıcıya girdi olarak verilmiştir. Sınıflandırma performansı 5-katlı çapraz doğrulama yöntemiyle F1-skoru kullanılarak ölçülmüştür. Deneylerde gövdelemenin çok sözcüklü ifade çıkarımında performansı geliştirmediği görülmüştür. Bununla beraber, çok sözcüklü ifade tespiti deneylerinde diğer yöntemlerden sürekli olarak üstün olan bir sınıflandırma yöntemi olmadığı gözlenmiştir.

Anahtar Kelimeler: Çok Sözcüklü İfadeler, Sözcük Temsili, Türkçe.

ACKNOWLEDGEMENT

I would like to take the opportunity to thank some people who supported me during the process of this thesis.

First of all, I would like to thank to my supervisor Assoc. Prof. Dr. Senem KUMOVA METİN who gave me the opportunity to work with her, for her excellent guidance, encouragement and patience.

I am very thank full to all my friends, especially Mustafa Kemal BİNLİ, Cem KUTLUER and Dünya KILAVUZ for their trust and unconditional support all the time.

Lastly, I would like to thank to my mother Şafak TAŞÇIOĞLU and my father Nezih TAŞÇIOĞLU for always being with me and motivating me with their endless support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The term "multiword expression"(MWE) refers to combination of two or more words that presents to say things in a traditional way [1]. In earlier studies the term "collocation" is used as an alternative for "MWE". The idea of MWE has been determined firstly in 1967 by J.R.Firth [2]. He claims that a word can be comprehended by group it preserves. In his later work, he claims "collocations of a given word are statements of the habitual or customary places of that word". Subsequently, Sinclair, determined collocation as occurrence of two or more words inside a short interspace of each other in text [3]. Conversely, Hoey determined a collocation that comprises of two or more lexical element jointly with a possibility which cannot be constructed randomly [4].

It is difficult to describe questions on MWE such as "what is MWE?" and "what is not MWE?". Because, MWE does not comprise of body of rules. However, in earlier studies in the literature, some common properties that are kept by all MWEs are determined. In study of Kumova and Karaoğlan [5], common properties are collected under 4 different categories. The first one is that collocations are frequent. Frequency is the easiest feature to calculate while differentiating MWEs from combinations of other words. The second one is that collocations are uncertain and language specific. There are not specific rules that determine which words are juxtaposed and how a word selects a specific word to produce collocation. For instance, "strong" is a common MWE with "coffee" in English. However, there is no distinct explaining why "coffee" does not collocate with "powerful". Also, MWEs can change in different languages rely on the cultural or social behaviours of native speakers. For example, in Turkish "strong coffee" is "sert kahve", but when it is translated word to word it means "hard coffee". The third title is that collocations produce a block in language. When sense or meaning integrity is considered in NLP (natural language processing), collocations can be accepted as a single word that has a specific meaning. The last title is that collocations are field dependent. For instance, "hard disk" does not mean that a disk is hard. It is a hardware component of computers which is used to store data. In this thesis, considering various properties of MWEs, four types of word combinations are accepted to be MWEs. Although MWEs consist of two or more words, in this thesis we focused on sequential two-word collocations (bigrams). The types of word combinations that are accepted as MWEs are given below:

- Compound verbs: Compound verbs are the first type of MWEs in this study. There exists several verbs that are built by combination of different words in Turkish as well as other languages. For example "aklını çelmek" is a commonly used compound verb that has an idiomatic meaning which refers to "dissuade" in English.

- Area specific terms: The second type of MWEs are the area specific terms that are well known and commonly used by area experts but rarely known especially by non-native speakers.

- Combination of words and conjuctions: There exists conjuctions that involve multiple words in most of the languages "ne kadar", "yok artık" are the examples of word combination in Turkish that reside in this type of MWEs.

- Personal names, named entities, abbreviations, job titles: The last type of MWEs include personal names such as "Mustafa Kemal", "Fatih Portakal"; abbreviations such as "Prof. Dr.", "Yrd. Doç." and job titles such as "genel müdür", "bilgisayar mühendisi".

In the literature, several methods are used to detect MWEs such as rule based, statistical and linguistic methods. Most of earlier studies in literature used statistical methods such as frequency of occurrence, mutual information, hypothesis testing. Frequency of occurrence method is the earliest and simplest technique to extract MWEs. In this method, the occurrence frequency of words or co-occurrence frequency of words indicates whether the combination of word is a MWE or not. Commonly, in this method, word combinations are sorted by their frequency based scores and MWE candidate list is created. Simply, the most frequent word combinations are assumed as MWEs. In mutual information, the mutual dependency of two words is measured. If mutual dependency is measured as 0 the word combination is accepted as non MWE. On the other hand, if the dependency result is far from 0, word combination is accepted to be a true MWE. In hypothesis testing, it is required to verify that common occurrence of the words is more than chance for deciding whether a combination of words is a MWE or not. Testing the hypothesis of independence is the popular approach demonstrating the dependence between words. Hypothesis testing techniques attempt to decline null hypothesis which states that words in combination occur independent of each other. Dunning's log-likelihood test, t-test, and chi-square (x^2) are examples to hypothesis testing techniques.

In recent natural language processing studies, researchers use word vectors instead of frequency based information. Word vectors are accepted to hold semantic information about the regarding word/word combination. The vector of a given word is built up considering the words that commonly co-occur with the regarding word. In this approach it is accepted that the meaning of a word is distributed over the neighbouring words. This is why, simply the vectors are built up by some type of frequency information of neighbouring words. There are several different approaches to represent the word/word combination with vectors. The most populars are Word2Vec, GloVe, Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA).

In this thesis, we aim to detect MWE by using word vector representations. Firstly, IMST (ITU-METU-Sabancı Treebank) [[6], [7], [8]] corpus is used to collect positive and negative MWE samples (word combinations). Secondly, we built up vectors of word combinations and composing words both in surface and stemmed forms using Bilkent [9], Leipzig [10] and Wikipedia. Five different methods, Word2Vec (Continuous Bag of Words, Skip Gram), GloVe, LSA, LDA, are employed to built vectors. The resulting vectors are used to construct the final vector of MWE candidate. And lastly, accepting MWE detection as a classification problem, MWE candidate vectors are given as input to ten different classifiers. The classification performance is evaluated by well-recognized F1 measure and the experimental results are reported.

This thesis is organised as follows. In chapter 2 the methodology is explained. This section covers the word vector representation methods, dataset preprocessing steps and classification methods. In section 3, related work on MWE detection is given. Chapter 4 involves the experimental results and chapter 5 is conclusion.

Chapter 2

Methodology

In this thesis, we accept that vector representation of a word (and/or word combination) refers to the meaning that it conveys. As a result, if several words combine to convey a different meaning (to build up a MWE), the vector representation of the word combination must be dissimilar to the individual representations of composing words. In our experiments, the average vector representation of composing words is subtracted from the vector representation of the word combination. And the resulting vector is considered as a measure of dissimilarity. This vector is accepted to be the representation of MWE candidate (from now on it will be named as MWE candidate vector) and it is given as an input to the classifier in order to label the candidate as MWE or non-MWE.
The following tasks are performed respectively in our experiments:

- Positive samples of MWEs are collected from the data set. This data set includes manually labelled MWEs. In section 2.2, the details of the data set and some statistics are given.
- The set of negative MWE samples is built by selecting most frequently occuring word combinations in the labelled data set.
- Vector representations of word combinations (both positive and negative samples) and composing words are built using Bilkent [9], Leipzig [10] and Wikipedia. Five different methods (Continuous Bag of Words, Skip Gram,

Glove, Latent Semantic Analysis, Latent Dirichlet Allocation) are used to generate representations.

- Average vector representation of composing words are obtained by addition of composing word vectors and dividing this vector by two.
- MWE candidate representations (vectors) are generated by subtracting the average vector of composing words from the vector of word combination.
- The set of MWE candidates vectors are utilized in classification experiments where 5-fold cross validation is used. The performance of 10 classification methods are compared using averaged F1 scores.

In Figure 2.1, the flow-chart gives an overview to the general structure of the thesis.

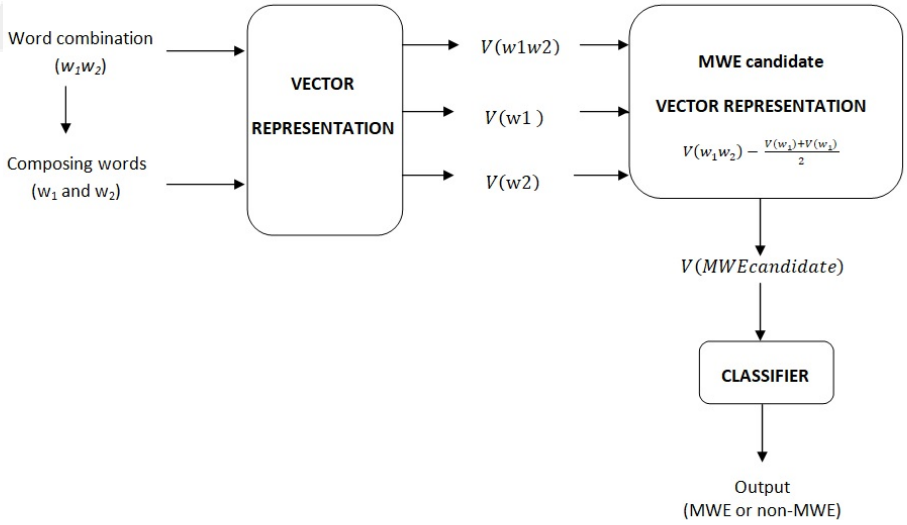


Figure 2.1: Overview of the thesis

In following subsections, word representation methods will be presented, the steps to preprocess dataset will be given, classification methods employed in this study will be briefly explained.

2.1 Methods of Word Representation

2.1.1 Word2Vec

Mikolov et al. proposed Word2Vec model in 2013. In order to learn word embeddings from raw text, this model uses computationally-efficient predictive model. This model is a simple neural network with only one hidden layer. The aim of neural networks is to arrange the weights in layers to decrease a loss function during training. In the proposed, Word2Vec method two models are presented. These are Continuous Bag of Words model-CBow and Skip Gram-sg model. These two models are similar in terms of their algorithms except that in the CBOW architecture, the model predicts target words from source context words. On the other hand, in the skip-gram architecture, the model does the opposite and predicts source context-words from the target words.

2.1.1.1 Architecture

Word2Vec model consists of three layers named as input layer, hidden layer and output layer. In Figure 2.2 the architecture of Word2Vec is demonstrated. The model takes a huge input vector, compresses it until creating a smaller dense vector then it outputs possibilities of target words. In other words, the input layer of Word2Vec is one-hot vector which has the same size with the vocabulary. In order to create one-hot vector, it is filled with zeros except the index of the input word. The hidden layer has the weights which are the word embeddings. This layer works as a lookup table and the output layer produces possibilities of target words from the vocabulary. In this layer a softmax activation function is executed. This function is below:

$$\alpha(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x^k}}$$

In order to maximize the probability of next words given the previous commonly a maximum likelihood principle is used in training. However, this is so

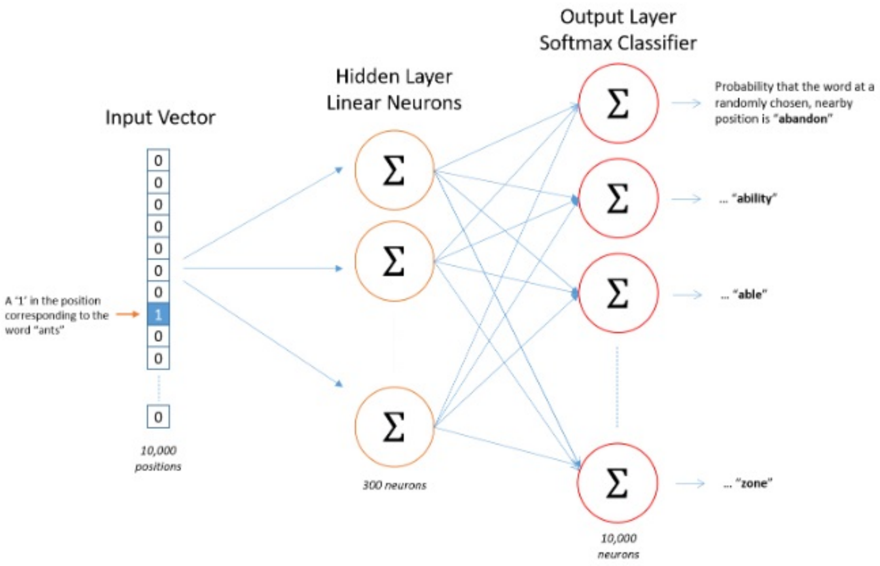


Figure 2.2: Architecture of Word2Vec [54]

computationally expensive for a large set of vocabulary. Because of this reason, Word2Vec model use hierarchical softmax and negative sampling in training and commonly negative sampling is preferred.

2.1.1.2 Continuous Bag of Words (CBoW)

Continuous Bag of Words model predicts a center word from surrounding text. Briefly, in this model a context window of words is determined for the target word each neighbouring word in the window is given to the neural network as an input. Then neural network is expected to the target word. In Figure 2.3 the neural network representation is given. In this figure, the term x_{ik} refers to a word in context window.

To exemplify, assume that "bir yandan da hiç konuşmak istemiyor" is our sentence as given in Figure 2.4. If window size = 1 then it means that each target word will be predicted by preceding and following word. The algorithm begins with the first word of the sentence as a center/target word. The first word in our example is "bir". It is predicted by its following word "yandan". Then the algorithm shifts to second word of sentence and the words "bir" and "da" inputs

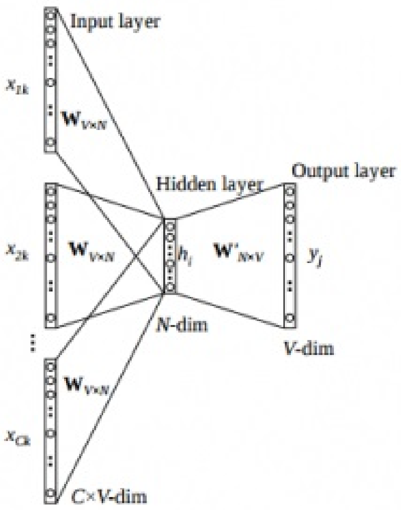
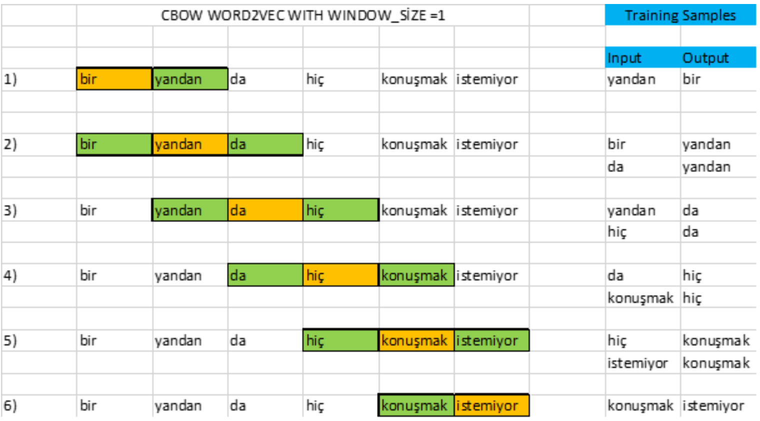


Figure 2.3: Neural Network Representation of CBoW

to the neural network to predict the second word "yandan" and it continuous till the end of the sentence.



	CBOW WORD2VEC WITH WINDOW_SİZE =1							Training Samples	
								Input	Output
1)	bir	yandan	da	hiç	konuşmak	istemiyor		yandan	bir
2)	bir	yandan	da	hiç	konuşmak	istemiyor		bir	yandan
								da	yandan
3)	bir	yandan	da	hiç	konuşmak	istemiyor		yandan	da
								hiç	da
4)	bir	yandan	da	hiç	konuşmak	istemiyor		da	hiç
								konuşmak	hiç
5)	bir	yandan	da	hiç	konuşmak	istemiyor		hiç	konuşmak
								istemiyor	konuşmak
6)	bir	yandan	da	hiç	konuşmak	istemiyor		konuşmak	istemiyor

Figure 2.4: CBOW Sample [55]

2.1.1.3 Skip-gram (SG)

Skip-gram model predicts surrounding context words given a center word. It is largely the same with CBoW except that the input one hot vector which is center word. In Figure 2.5, the neural network of SG is given.

In Figure 2.6, an example sentence and the contents of input and output layers when SG algorithm is executed is given. Briefly, the algorithm takes the center word as an input and tries to predict context words. Firstly, the word "bir" is the center word and try to predict the word "yandan" by using center word. Then, it shifts the center word to right. Now, the center word is "yandan". It tries to predict "bir" and "da" by using "yandan" and it continues like this until the end of sentence.

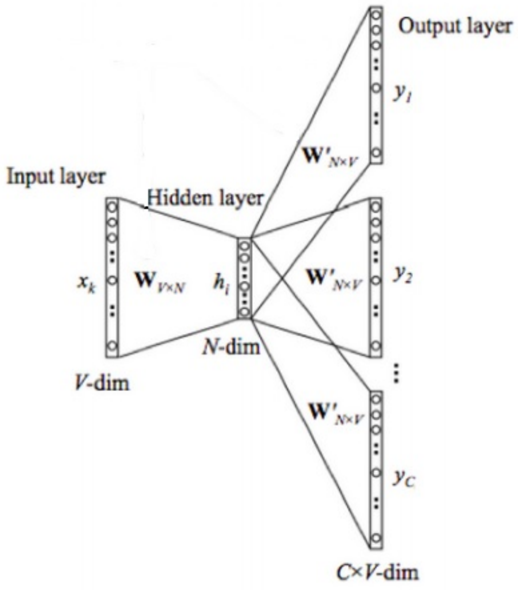
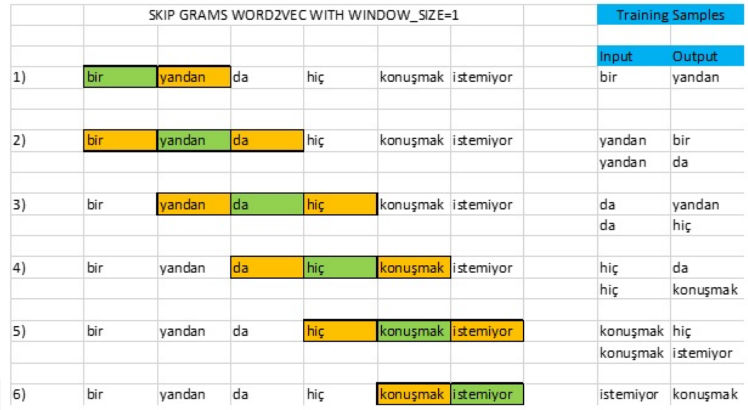


Figure 2.5: Neural Network Representation of SG



	SKIP GRAMS WORD2VEC WITH WINDOW_SIZE=1					Training Samples	
						Input	Output
1)	bir	yandan	da	hiç	konuşmak	istemiyor	bir	yandan
2)	bir	yandan	da	hiç	konuşmak	istemiyor	yandan	bir
							yandan	da
3)	bir	yandan	da	hiç	konuşmak	istemiyor	da	yandan
							da	hiç
4)	bir	yandan	da	hiç	konuşmak	istemiyor	hiç	da
							hiç	konuşmak
5)	bir	yandan	da	hiç	konuşmak	istemiyor	konuşmak	hiç
							konuşmak	istemiyor
6)	bir	yandan	da	hiç	konuşmak	istemiyor	istemiyor	konuşmak

Figure 2.6: SG Sample [55]

2.1.2 GloVe

GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm which developed by Stanford University to generate vector representations for words. The aim of GloVe is to produce word vectors which find the "meaning in vector space" by using statistics of global count. Distinctly from Word2Vec, Glove learns based upon a co-occurrence matrix and trains vectors thus their differences estimate co-occurrance ratios. In GloVe model two main methods are used these are global matrix factorization and local context window. Local context windows are CBoW and SG. As mentioned before, these two methods require a predefined window of words. The global matrix factorization is used to reduce large term frequency matrices in Latent Semantic Analysis. And also, this method is used in GloVe to include global frequency information in order to build up word vectors. In GloVe model, instead of co-occurrence probabilities the ratio of co-occurrence probabilities are used. Further details on GloVe model can be found in[11].

2.1.3 Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) or Latent Semantic Indexing (LSI) is a method in NLP to analyse relationships between a group of documents and to produce a group of concepts associated to these documents and terms. It supposes that words which are close in terms of meaning are locating in similar parts of text. In this method, a matrix is built in which rows are the unique words and the columns are the paragraphs. This matrix is such a huge that it can not be processed in its raw form this is why the mathematical technique singular value decomposition-SVD is used to decrease the number of rows. This reduction is actually removing the words that are not important, in other words does not convey information in text. In the following subsection the details on derivation of LSA are given.

2.1.3.1 Derivation of LSA

Assume that X be a matrix where element i, j defines the occurrent of term i in document j:

$$X = \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix}$$

Each row in this matrix indicates a vector corresponding to a term and its affiliation to each document:

$$tj^T = \begin{bmatrix} x_{1,1} & \dots & x_{i,j} & \dots & x_{i,n} \end{bmatrix}$$

Similarly each column in this matrix indicates a vector corresponding to a document and its affiliation to each term.

$$dj = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{i,j} \\ \vdots \\ x_{m,j} \end{bmatrix}$$

The dot product $t_i^T t_p$ between two term vectors yields the correlation between the terms over the set of documents and the matrix product XX^T contains all these dot products. Element $(i, p)\, or\, (p, i)$ contains the dot product $t_i^T t_p \left(= t_p^T t_i\right)$.In the same way, the matrix $X^T X$ includes dot products between all the documents vectors and their correlation over the terms: $d_j^T dq = d_q^T dj$ According to linear algebra, there is a decomposition of X such that U and V are orthogonal matrices and Σ is diagonal matrix. This is entitled Singular Value Decomposition (SVD):

$$X = U\Sigma V^T$$

The matrix products give the correlation between terms and documents:

$$\begin{aligned} XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V^{T^T}\Sigma^T U^T) = U\Sigma V^T V\Sigma^T U^T \\ = U\Sigma\Sigma^T U^T = U\Sigma^2 U^T \end{aligned}$$

$$\begin{aligned} X^T X = (U\Sigma V^T)^T(U\Sigma V^T) = (V^{T^T}\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T \\ = V\Sigma^T\Sigma V^T = V\Sigma^2 V^T \end{aligned}$$

Since $\Sigma\Sigma^T$ and $\Sigma^T\Sigma$ are diagonal that U contains the eigenvector of XX^T while V is eigenvectors of $\Sigma^T\Sigma$. Both of the products own the same non-zero eigenvalues, given by the non-zero entries of $\Sigma\Sigma^T$ or by the non-zero entries of $\Sigma^T\Sigma$. Then the decomposition:

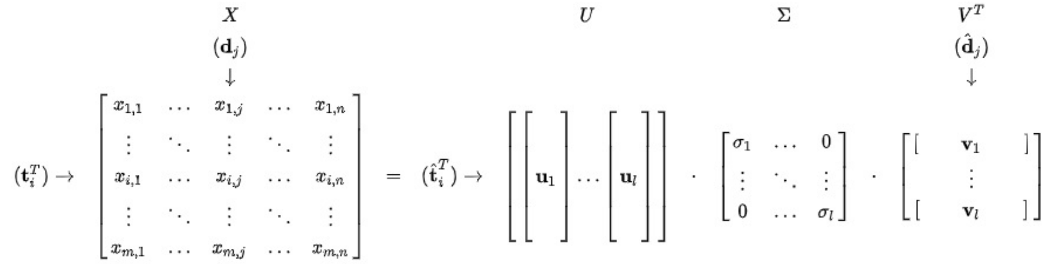

X
(d_j)
U
Σ
V^T
(d̂_j)
(t_i^T) →
(t̂_i^T) →
x_{1,1} … x_{1,j} … x_{1,n}
x_{i,1} … x_{i,j} … x_{i,n}
x_{m,1} … x_{m,j} … x_{m,n}
u_1 … u_l
σ_1 … 0
0 … σ_l
v_1
v_l

Figure 2.7: Singular Value Decomposition from Wikipedia

The values $\sigma_1, \ldots, \sigma_l$ are singular values, and $u_1, \ldots, u_l$ and $v_1, \ldots, v_l$ the left and right singular vectors. Notice that the only part of U contributes to t_i is the i'th row. Assume that $\hat{t}_i^T$ row vector. Similarly, the only part of V^T that makes contribution to d_j is the j'th column, $\hat{d}_j$. When the f largest singular values, and their corresponding singular vectors are selected from U and V, it gives the result of the rank f approximation to X the smallest error. As a result, the term and document vectors create *semantic space*. The "term" vector $\hat{t}_i^T$ which indicates the row has f items that indicate lower-dimensional space dimensions. These new dimensions are lower-dimensional approximation of the higher-dimensional space. Similarly, the "document" vector $\hat{d}_j$ is an approximation in this lower-dimensional space. Then the approximation is:

$$X_f = U_f \Sigma_f V_f^T$$

2.1.4 Latent Dirichlet Allocation (LDA)

Latent dirichlet allocation is a technique that is used widely in natural language processing for topic modeling. It is a “generative probabilistic model” for a collection of documents that are represented as combination of latent topics where each topic is characterised by a distribution over the words. In other words, it tries to obtain some “topics” that represent the collection of documents.

In LDA, topics are assumed to be specified before any information is generated and documents are represented as combination of topics that distribute the words with certain probabilities. As a result, the presence of a word is accepted to be indicator of a specific topic in document. The probabilistic model estimated by LDA consists of two matrices. The first matrix shows the probability or chance of selecting a particular word when sampling a particular topic. The second one describes the chance of selecting a particular topic when sampling a particular document. In LDA, the below algorithm is executed to generate documents:

- Build a unique set of words.
- Decide on the number of documents that will be generated.
- Decide on the number of words per document (sample from a Poisson distribution).
- Decide on the number of topics you want.

- Pick a number between not-zero and positive infinity and call it alpha.
- Pick a number between not-zero and positive infinity and call it beta.
- Build the 'words-versus-topics' table. For each column, draw a sample from a Dirichlet distribution using beta as the input. Each sample will fill out each column in the table, sum to one, and give the probability of each part per topic.
- Build the 'documents-versus-topics' table. For each row, draw a sample from a Dirichlet distribution using alpha as the input. Each sample will fill out each row in the table, sum to one, and give the probability of each topic per document.
- Build the actual documents. For each document,
    1. look up its row in the 'documents-versus-topics' table
    2. sample a topic based on the probabilities in the row
    3. go to the 'words-versus-topics' table
    4. look up the topic sampled
    5. sample a part based on the probabilities in the column
    6. repeat from step 2 until you've reached how many words this document was set to have

2.2 Dataset Preprocessing

In this study, several textual resources are employed and several preprocessing tasks are performed prior to the experiments. Firstly, a modified version of METU-Sabancı treebank [[6], [7], [8]] known as ITU-METU-Sabancı Treebank (IMST) that is splitted into multiple units and manually tagged by ITU NLP group is utilized to build up sets of positive and negative MWE samples. IMST corpus involves 5635 number of sentences and a total of 56422 words. There exists 2040 number of MWE labeled word combinations (both bigrams and trigrams) in regarding corpus. In preprocessing, punctuations are removed and all letters

	Vector representations				
Corpus	LDA	LSA	GLOVE	CBOW	SG
Combined corpus - surface	-	-	+	+	+
Combined corpus - stem	-	-	+	+	+
Bilkent - Surface	+	+	+	+	+
Bilkent - Stem	+	+	+	+	+

Table 2.1: Available vector representations in the experiments

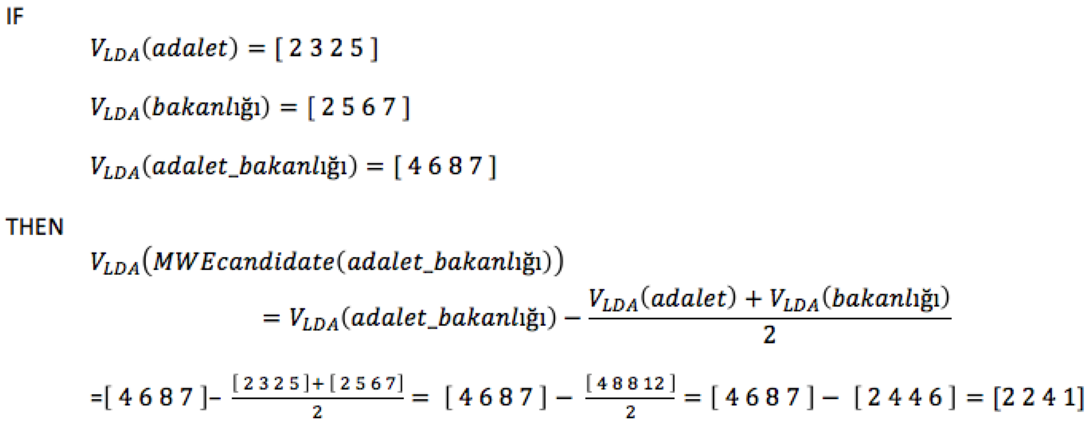
are converted to lowercase in corpus. Following, IMST corpus is splitted into bigrams in order to collect MWE samples that are built up with two sequential words. Occurrence frequencies of bigrams are determined and frequently occurring bigrams are accepted as negative samples of multiword expressions. Positive samples that are labeled as true MWEs in treebank are excluded from the set of negative samples. As a result, 1349 positive and 1349 negative unique bigrams are obtained from the surface form of the corpus. The same procedure is applied after the corpus is stemmed using Turkish Stemmer for Python [12]. From the resulting stemmed version of corpus, 1247 number of positive, 1304 number of negative MWE samples are collected following the similar steps.

Vector representations of samples and composing words are obtained from Bilkent corpus and a merged version of Bilkent [9], Leipzig corpora [10] and Turkish Wikipedia articles (downloaded on 03.12.2018). The sizes of corpora in terms of tokens are 719665 and 63632928 respectively for surface forms of Bilkent and combined corpora. The regarding corpora are also stemmed and a stemmed version for both corpora are utilized to generate vectors for stemmed MWE samples.

Due to the limited resources, combined corpus was unable to be used to build up vectors by Latent Semantic Analysis and Latent Dirichlet allocation. The resulting available vector representations in this study are given in Table 2.1.

Each vector representation in our data set is actually a fixed-size array of numbers that is accepted to hold the meaning of the regarding unigram or bigram. In our experiments vector size for LDA and LSA methods are set to fifty; for GloVe, CBoW and SG the vector sizes are 100. For example, in order to build up the vector representation of MWE candidate *adalet bakanlığı*, vectors of *adalet* and *bakanlığı* are summed up and divided by two to obtain the average vector. And this average vector is subtracted from the vector of bigram *adalet bakanlığı* to obtain the vector of MWE candidate. In order to generate the vector of bigram *adalet bakanlığı*, underscore (_) character is put between each sequential occurrence of unigrams *adalet* and *bakanlığı*.

In Figure 2.8, the steps to generate the LDA vector for MWE candidate "adalet_bakanlığı" is given as an example. In this figure, $V_{LDA}(X)$ represents the vector that is produced by LDA method for the unit X. X may be either a composing word or the word combination.



IF

$$V_{LDA}(adalet) = [\,2\;3\;2\;5\,]$$

$$V_{LDA}(bakanlığı) = [\,2\;5\;6\;7\,]$$

$$V_{LDA}(adalet\_bakanlığı) = [\,4\;6\;8\;7\,]$$

THEN

$$V_{LDA}(MWEcandidate(adalet\_bakanlığı)) = V_{LDA}(adalet\_bakanlığı) - \frac{V_{LDA}(adalet) + V_{LDA}(bakanlığı)}{2}$$

$$=[\,4\;6\;8\;7\,] - \frac{[\,2\;3\;2\;5\,]+[\,2\;5\;6\;7\,]}{2} = [\,4\;6\;8\;7\,] - \frac{[\,4\;8\;8\;12\,]}{2} = [\,4\;6\;8\;7\,] - [\,2\;4\;4\;6\,] = [2\;2\;4\;1]$$

Figure 2.8: Generating the LDA vector

In this study, it is examined that while building up the vector representations of a limited number of MWEs, since the composing unigrams and/or the candidate itself does not occur in corpus, it is not possible to generate the vector representation of regarding MWE candidate. As a result we excluded the MWE candidate from the experiments if any of its composing words or MWE candidate occurs less than 5 times in our corpus.

The vector representations are given as inputs to 10 different supervised machine learning algorithms as the next step. We employed Weka machine learning tool [13] to run classification algorithms with 5-fold cross validation in order to classify each sample as either MWE or non-MWE. In 5-fold cross validation, the data set is splitted into 5 folds. In each iteration, one fold is used in testing and remaining four folds are used in training. As a result the training/testing is performed five times and the performance is evaluated. The average classification performance is presented.

In our experiments, we employed well-known classification measures of precision, recall and F1 for evaluation. Precision and recall are

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

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

where TP refers to the number of samples that are both labeled (expected) as MWE and classified (observed) as MWE, FP is the number of samples which are classified as MWEs falsely. In other words, they are the negative samples that are observed as true MWEs by classification method. FN is the number of samples that are classified as non-MWEs but are true MWEs in manually labeled set.

F1 measure is the weighted harmonic mean of the precision and recall measures. It is calculated as:

$$F1Score = 2 * (Recall * Precision) / (Recall + Precision)$$

2.3 Classification Methods

2.3.1 Naive Bayes

Naive Bayes classifier is a classification method that bases on Bayes Theorem. It is known as one of the simplest and fastest algorithms. In this classifier, features/inputs are assumed to be independent of each other and all features are dependent to the classification label. Based on this assumption, the sample is assigned to the related class by calculating the multiplication of its conditional probability score to each class. The details on Naive Bayes classifier that is used in this study is given in study of John and Langley [14].

2.3.2 BayesNet (Bayesian Network)

Bayesian network, also known as belief network, is a member of the family of probabilistic graph models [15]. Bayesian network has a graphical model structure known as “directed acyclic graph” and it permits the representation of common probability distributions actively and effectively [16]. The nodes used in these graphs represent features and the edges represent probabilistic conditional dependencies. These networks provide an efficient representation of multivariate probability distribution of a set of random variables and allow various calculations [17].

A set of nodes on the path from a node in Bayesian Networks is described as “descendants”, a set of nodes located on the path to a node are described as “ancestors”. Directed a cyclic graph prevents a node being descendant or ancestor of itself.

In our study, standard estimators presented in WEKA tool and searching algorithms are preferred. In other words, for calculation of conditional probability values simple estimator is used and k2, a hill climbing algorithm, is used as searching algorithm. Detailed information about Bayesian Networks and related algorithm is presented in study of Bouckaert [18].

2.3.3 SMO (Sequential Minimal Optimization)

Sequential Minimal Optimization (SMO) is simply a support vector machine algorithm in order to perform regression analysis [[19], [20]]. In support vector machines, briefly the decision boundary that split the two groups of samples on the same plane is to be determined. Basically, a boundary that has the maximum distance to samples of both groups is found and a complete separation is satisfied. The class of new sample is determined based on this boundary. In generalized form of SVMs, they are the machines that may be built up a hyper-plane or a set of hyperplanes. These hyper-planes are used in classification or regression tasks [[20], [21]].

In this thesis, we employ WEKA implementation of SMO algorithm, which is actually a support vector regression algorithm. In sequential minimal optimization algorithm, multi dimensional problem is divided into two-dimensional problems and is solved analytically. This is why, it is accepted to be a better solution compared to relatively harder SVM problems [22].

2.3.4 Logistic Regression

Logistic regression is a method that estimates the class/value of dependent variable by values of continuous or categorical independent features [23]. Although, the method is similar to other multivariate methods, discriminant analysis etc., some assumptions are not valid in this method. For instance, in this method independent features are required to be normally distributed, and homogeneous variance and covariance of classes are not valid/required.

In the logic regression method, variable z that is called logit and includes the contribution of all features is given as an input. Variable z and the output f(z) of the system are stated as below:

$$z = b_1x_14b_2x_2 + b_3x_3 + ... + b_kx_k + c(1)$$

$$f(z) = \frac{e^z}{1 - e^z}$$

In equation 4, given bi values represent the coefficients of features. f(z) produces a value in the range of [0,1] and indicates that what should be the label of the regarding instance/sample.

In our study, WEKA implementation of a logistic regression model that includes ridge estimators is used. This model is based on the method proposed in Le Cessie and Van Houwelingen [24]. Detailed mathematical background information is presented in [24].

2.3.5 Voted Perceptron

Voted perceptron is a classical perceptron algorithm in which a version of Helmbold and Warmuth's leave-one-out algorithm is employed in batch learning [25]. The perceptron is mainly used to determine the prediction vector v where $y = sign(v.x)$ given x is the set of samples and y is the set of labels. The prediction vector is initialized to zero. Following, for each sample in training set prediction vector is used to predict the label. If the label is true for the sample, the vector does not change. If the label is false, then the vector is modified based on the rule $v = v + yx$. This modification is repeated till all the samples in the training set are labeled correctly. In leave-one-out algorithm, a list of r number of training samples is built and one unlabeled sample is added to the list. Following the training step, the label of the last sample is predicted. In Weka implementation of voted perceptron tool, majority vote algorithm and a vector weighting score that the number of iterations till the first incorrect prediction is used. Further details on voted perceptron algorithm may be found in [26].

2.3.6 K-nearest Neigbour

IBk classifier is an implementation of k- nearest algorithm in weka tool. In K-nearest neighbour classifier, positive and negative samples are classified based on class label of their k-nearest neighbours. For each sample, the labels of the nearest k samples are investigated and the most frequently observed class label is assigned to regarding sample. It is common to choose k as a small and an odd number (1, 3 or 5 etc) to break equality. For example, if k=1 the label of sample is determined according to the label of single nearest neighbour. The distance between sample and neighbour can be calculated by various distance measures like Euclidean, Manhattan Distance etc. In our study, values of k are set to 5 and 10. In other words, the first nearest, top 5 and top 10 neighbours are examined and the new sample is.

2.3.7 AdaboostM1

Adaptive Boosting, Adaboost, is a machine learning meta-algorithm proposed by Freund and Schapire [27]. This method may be used with other learning algorithms to improve performance of regarding algorithms. The basis of the method lies in accepting the weak classifiers that performances are slightly better than random prediction as experts and making a common decision by combining several numbers of predictions [28]. In this sense, Adaboost is used to reduce the margin of error of weak classifiers [27]. The Adaboost algorithm used in the solution of binary classification problems is called AdaboostM1 and the one used in the solution of multiple classification problems is called AdaboostM2 [29].

In our study, "AdaboostM1" algorithm implemented in WEKA tool is used as recommended by Freund and Schapire[27]. The weak classifier underlying the method is a one-level decision tree called decision stump [30]. In this decision tree, the nodes that are connected to the root node are terminal nodes/leaf nodes. Each decision root decides on a single feature.

2.3.8 OneR (One Rule)

One Rule (OneR) is a classifier algorithm that is simple but efficient and easy to interpret. This classifier creates a rule for each feature (input) in dataset. In this algorithm, the rule that has the lowest number of errors is assigned as “one rule”. Simply, in order to determine the rules that belong to the features, the frequency table that shows the amount true/false classification for each feature is created. By this table, the feature that generates the lowest error and the relating rule is determined [31].

2.3.9 J48 (C4.5)

J48 classifier is the implementation of C4.5 decision tree in WEKA tool. Although C4.5 tree is similar to ID3 decision tree (Iterative Dichotomiser 3), the main difference between them is that ID3 uses information gain while C4.5 uses gain ratio [32]. On the other hand, different from ID3 tree C4.5 tree can be pruned. In C.4.5 algorithm, in order to create a new branch of the tree, the gain ratio value of the node is considered. After that, a sub-list is created under the new decision node and sub-decision trees are constructed. The unnecessary/useless branches are removed by pruning in order to narrow down the decision space by decreasing the size of tree.

2.3.10 Random Forest

Random Forest algorithm is based on decision trees. In this algorithm, training set is splitted into some number of sub-training sets. Each sub-training set is trained and their decisions are combined [34]. By using this method, more than one classifier is obtained and the classification votes (produced labels) of these classifiers are used in classification of samples. In order to build the trees independent of each other instead of selecting best classifying features (inputs), the features are randomly selected. In random forests, the trees are not pruned [[33], [34]]. When the method is compared with the similar methods it is faster and better in dealing with over fitting. Also, it is possible to produce desired number of trees [35].

Chapter 3

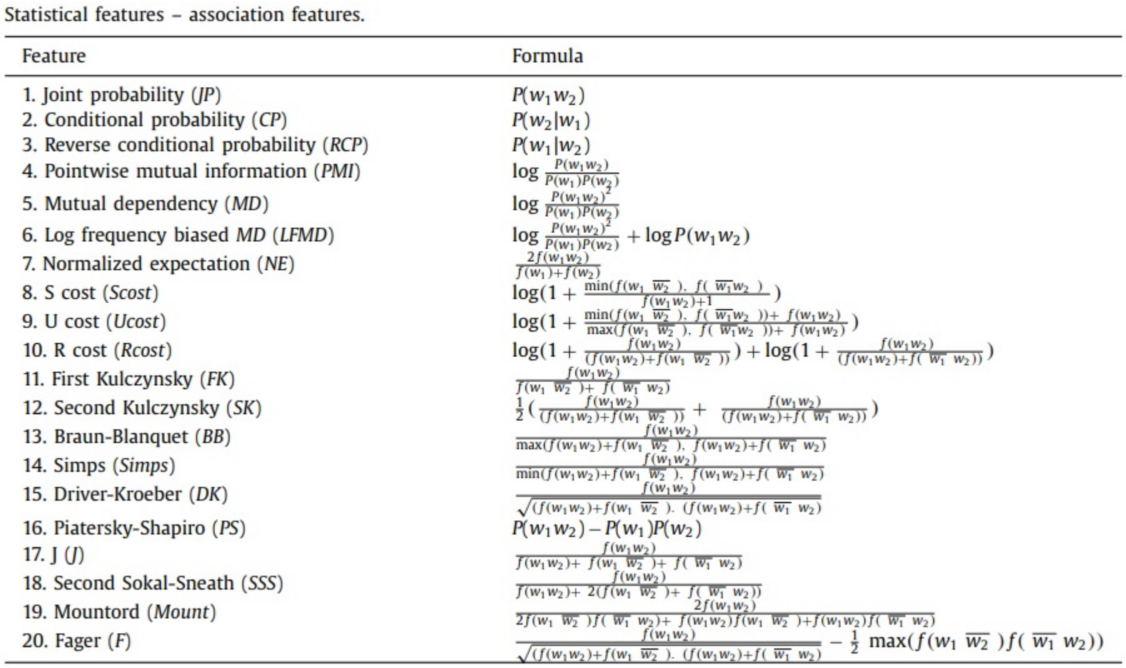
Literature Review

The multiword expressions (MWE) consist of associating two or more words that represent to say things in traditional way [1]. In some previous studies, MWE is also named as "collocation". MWE has been identified in 1967 by J.R. Firth [2]. He claims that a word can be comprehended by the group it preserves. In his later work, he claims "collocations of a given word are statements of the habitual or customary places of that word". Subsequently, Sinclair, determined collocation as occurrence of two or more words inside a short interspace of each other in text [3]. Conversely, Hoey determined a collocation as two or more lexical elements that join in text with a possibility that cannot be accepted as random [4]. In this thesis, we accepted MWE and collocation as similar notions as mentioned commonly in the literature.

In previous studies in which MWE detection is accepted to be a binary decision task where the output is either MWE or non-MWE, the detection/extraction methods are classified into three major groups as rule-based, linguistic and statistical techniques. In rule-based techniques, a group of rules is predetermined and it is necessary to fire up the rules one by one to label MWEs. Due to this high number of rules, rule-based techniques have high complexities compared to other techniques. For example, Oflazer et al. [36] presented a rule-based MWE extraction processor for Turkish where there exists 1100 number of rules to assign the given candidate as MWE. Simply, the presence of some statistical and linguistically motivated features in a candidate results with the MWE decision.

Tsvetkov and Wintner [37] proposed a study where linguistic features are extracted from text to detect MWEs. Some of the linguistic features that are employed in their study are partial variety in surface form, frozen form, hapax legonema. In this study, instead of building up MWE rules, feature scores are given as input to the classifiers and binary classification methods are executed. In Sarıkas's study [38], different explanations and instances of MWEs in Turkish are presented and the relation between linguistic properties and the MWEs are discussed. In Kumova Metin [39], seven linguistic features are determined to extract MWEs in Turkish considering the language-specific properties. These features are partial variety in surface form, orthographical variety, frozen form, hapax-fossil, and the ratio of upper case letters, the suffix sequence, named entity words.

In Kumova Metin [39], statistical methods are presented as features that employ mainly occurrence frequencies of MWE candidates and their composing words. Statistical features are categorized in two groups as association features (association measures) and term-hood features. Association features grades the level of association between the words [40]. In other words, they measure the strength association between the words in collocation/MWE. In this approach, it is assumed that the words that co-occur frequently in text are closer to be true MWEs. For example, in Bouma [41] based on this assumption, mutual information and point-wise information which are well-recognized measures in different areas are proposed to be used in MWE extraction. In Table 3.1, a list of association measures given in Kumova Metin [39] are presented. In Table 3.1, the first and the second words are denoted by w_1 and w_2 in order for MWE candidate. The probability of co-occurrence of two words is denoted by $P\left(w_1 w_2\right)$ and $P\left(w_1\right)$ and $P\left(w_2\right)$ presents the occurrence probabilities of the first and the second words. $P\left(w_i|w_j\right)$ represents the conditional occurrence probability of w_i given that the word w_j is observed. Occurrence frequency of $w_1 w_2$, w_1 and w_2 are denoted by $f\left(w_1 w_2\right)$, $f\left(w_1\right)$, $f\left(w_2\right)$.

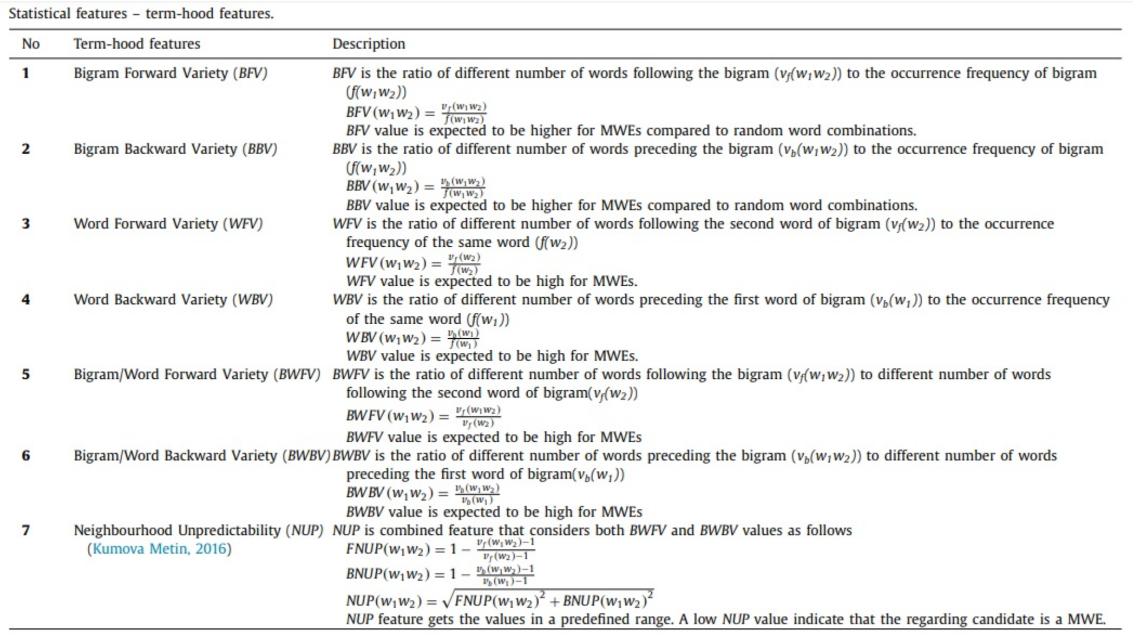


Statistical features – association features.

Feature	Formula
1. Joint probability (*JP*)	$P(w_1w_2)$
2. Conditional probability (*CP*)	$P(w_2 \mid w_1)$
3. Reverse conditional probability (*RCP*)	$P(w_1 \mid w_2)$
4. Pointwise mutual information (*PMI*)	$\log \frac{P(w_1w_2)}{P(w_1)P(w_2)}$
5. Mutual dependency (*MD*)	$\log \frac{P(w_1w_2)^2}{P(w_1)P(w_2)}$
6. Log frequency biased *MD* (*LFMD*)	$\log \frac{P(w_1w_2)^2}{P(w_1)P(w_2)} + \log P(w_1w_2)$
7. Normalized expectation (*NE*)	$\frac{2f(w_1w_2)}{f(w_1)+f(w_2)}$
8. S cost (*Scost*)	$\log(1+\frac{\min(f(w_1\overline{w_2}),\ f(\overline{w_1}w_2)}{f(w_1w_2)+1})$
9. U cost (*Ucost*)	$\log(1+\frac{\min(f(w_1\overline{w_2}),\ f(\overline{w_1}w_2))+f(w_1w_2)}{\max(f(w_1\overline{w_2}),\ f(\overline{w_1}w_2))+f(w_1w_2)})$
10. R cost (*Rcost*)	$\log(1+\frac{f(w_1w_2)}{(f(w_1w_2)+f(w_1\overline{w_2}))})+\log(1+\frac{f(w_1w_2)}{(f(w_1w_2)+f(\overline{w_1}w_2))})$
11. First Kulczynsky (*FK*)	$\frac{f(w_1w_2)}{f(w_1\overline{w_2})+f(\overline{w_1}w_2)}$
12. Second Kulczynsky (*SK*)	$\frac{1}{2}(\frac{f(w_1w_2)}{(f(w_1w_2)+f(w_1\overline{w_2}))}+\frac{f(w_1w_2)}{(f(w_1w_2)+f(\overline{w_1}w_2))})$
13. Braun-Blanquet (*BB*)	$\frac{f(w_1w_2)}{\max(f(w_1w_2)+f(w_1\overline{w_2}),\ f(w_1w_2)+f(\overline{w_1}w_2)}$
14. Simps (*Simps*)	$\frac{f(w_1w_2)}{\min(f(w_1w_2)+f(w_1\overline{w_2}),\ f(w_1w_2)+f(\overline{w_1}w_2)}$
15. Driver-Kroeber (*DK*)	$\frac{f(w_1w_2)}{\sqrt{(f(w_1w_2)+f(w_1\overline{w_2}).\ (f(w_1w_2)+f(\overline{w_1}w_2)}}$
16. Piatersky-Shapiro (*PS*)	$P(w_1w_2)-P(w_1)P(w_2)$
17. J (*J*)	$\frac{f(w_1w_2)}{f(w_1w_2)+f(w_1\overline{w_2})+f(\overline{w_1}w_2)}$
18. Second Sokal-Sneath (*SSS*)	$\frac{f(w_1w_2)}{f(w_1w_2)+2(f(w_1\overline{w_2})+f(\overline{w_1}w_2))}$
19. Mountord (*Mount*)	$\frac{2f(w_1w_2)}{2f(w_1\overline{w_2})f(\overline{w_1}w_2)+f(w_1w_2)f(w_1\overline{w_2})+f(w_1w_2)f(\overline{w_1}w_2)}$
20. Fager (*F*)	$\frac{f(w_1w_2)}{\sqrt{(f(w_1w_2)+f(w_1\overline{w_2}).\ (f(w_1w_2)+f(\overline{w_1}w_2)}}-\frac{1}{2}\max(f(w_1\overline{w_2})f(\overline{w_1}w_2))$

Figure 3.1: Expert Systems With Applications 92 [39]

Though association features are successful in many detecting types of MWEs such as conjunctions, compound verbs, they fail in detecting multiword technical terms. Since technical terms are rarely used in corpora that involve texts in different topics, it is hard to detect the association between their composing words. This is why, term-hood features are presented in previous studies. In term-hood features, the strength of ties between composing words and the strength of ties between a composing word and its neighboring word other than the other composing word are measured. It is possible to name these as strengths of inner and outer ties. If inner ties are stronger than the outer ties, term-hood measures accept the regarding candidate as MWE, and non-MWE otherwise. In Table 3.2, a list of term-hood features given in Kumova Metin [39] are demonstrated.



Statistical features – term-hood features.

No	Term-hood features	Description
1	Bigram Forward Variety (*BFV*)	*BFV* is the ratio of different number of words following the bigram ($v_f(w_1w_2)$) to the occurrence frequency of bigram ($f(w_1w_2)$) $BFV(w_1w_2) = \frac{v_f(w_1w_2)}{f(w_1w_2)}$ *BFV* value is expected to be higher for MWEs compared to random word combinations.
2	Bigram Backward Variety (*BBV*)	*BBV* is the ratio of different number of words preceding the bigram ($v_b(w_1w_2)$) to the occurrence frequency of bigram ($f(w_1w_2)$) $BBV(w_1w_2) = \frac{v_b(w_1w_2)}{f(w_1w_2)}$ *BBV* value is expected to be higher for MWEs compared to random word combinations.
3	Word Forward Variety (*WFV*)	*WFV* is the ratio of different number of words following the second word of bigram ($v_f(w_2)$) to the occurrence frequency of the same word ($f(w_2)$) $WFV(w_1w_2) = \frac{v_f(w_2)}{f(w_2)}$ *WFV* value is expected to be high for MWEs.
4	Word Backward Variety (*WBV*)	*WBV* is the ratio of different number of words preceding the first word of bigram ($v_b(w_1)$) to the occurrence frequency of the same word ($f(w_1)$) $WBV(w_1w_2) = \frac{v_b(w_1)}{f(w_1)}$ *WBV* value is expected to be high for MWEs.
5	Bigram/Word Forward Variety (*BWFV*)	*BWFV* is the ratio of different number of words following the bigram ($v_f(w_1w_2)$) to different number of words following the second word of bigram($v_f(w_2)$) $BWFV(w_1w_2) = \frac{v_f(w_1w_2)}{v_f(w_2)}$ *BWFV* value is expected to be high for MWEs
6	Bigram/Word Backward Variety (*BWBV*)	*BWBV* is the ratio of different number of words preceding the bigram ($v_b(w_1w_2)$) to different number of words preceding the first word of bigram($v_b(w_1)$) $BWBV(w_1w_2) = \frac{v_b(w_1w_2)}{v_b(w_1)}$ *BWBV* value is expected to be high for MWEs
7	Neighbourhood Unpredictability (*NUP*) (Kumova Metin, 2016)	*NUP* is combined feature that considers both *BWFV* and *BWBV* values as follows $FNUP(w_1w_2) = 1 - \frac{v_f(w_1w_2)-1}{v_f(w_2)-1}$ $BNUP(w_1w_2) = 1 - \frac{v_b(w_1w_2)-1}{v_b(w_1)-1}$ $NUP(w_1w_2) = \sqrt{FNUP(w_1w_2)^2 + BNUP(w_1w_2)^2}$ *NUP* feature gets the values in a predefined range. A low *NUP* value indicate that the regarding candidate is a MWE.

Figure 3.2: Expert Systems With Applications 92 [39]

In addition to above mentioned studies, in the literature, there exists various number of different studies that propose methods other than the previously mentioned ones, to extract MWEs or to build up automatically labelled MWE sets or that present toolkits to detect MWEs in texts. Below some of examples to these studies are given.

In the study of Corderio et al. [42], a multiword expression toolkit is presented as an extension to mwetoolkit[43] where semantic compositionality scores are employed. These compositionality scores are obtained by word embeddings (e.g. CBOW, glove).

Kumova Metin [44] presented common-decision and comparison based co-training approaches in order to find how unlabeled data can be used to extend the dataset of labelled MWE training. In the study, the performances of the proposed approaches are compared with the standard co-training in Blum [45]. Statistical and linguistic properties are used as two separate views of the MWE dataset [46]. A group of tests is applied with different settings on a Turkish MWE dataset and ten different classification algorithms are used. In study of Metin and Karaoğlan [47], statistical methods such as hypothesis tests, occurrence frequency, pointwise mutual information are used to identify collocations automatically for Turkey Turkish corpus. The methods are evaluated by using F-measure.

The aim of Uymaz and Metin [48] is to provide an analysis of the performance changing of frequency based measures when corpus is exchanged with a dynamic and massive data source, the “ World Wide Web ”. In this study, in order to obtain web-based frequencies three different experiments are performed by using 20 frequency based metrics. In ranking, the MWE candidates are evaluated their tendency for being MWE. Secondly, a feature selection method determines the most successful frequency metrics. The last one is, MWE extraction is accepted as a classification problem. In order to present incorporated performance of frequency metrics during frequency is acquired from web eight supervised techniques are used.

In study of Kumova Metin [49], a measure based on the power of cohesive outer ties between the words for identifying MWEs in a corpus is proposed. It not only allows the detection of bigrams it also detects trigrams in the corpus when applied recursively.

Eren and Metin [50] explored the performance of vector space models (VSM) in finding of non-compositional expressions (e.g. idioms) for Turkish language. A data set that consists of 2229 uninterrupted two-word sequential combinations that is created from six different Turkish corpora is used. Three sets of five different VSMs are utilizes in the experiments. The results of experiments are evaluated by accuracy and F-measures.

Pedersen [51] proposed three systems that include distributional techniques of measuring semantic compositionality. These systems mentioned semantic compositionality as a collocation identification problem, assuming that strong collocates are minimally compositional.

Vecchi et al. [52] presented an approach to characterise the semantic deviation of complex expressions. In their study, vector based semantic space is used to explore features of “ adjective-noun complex expressions ”. In order to do that, they suggested various models to demonstrate compositionality degrees of adjective noun expressions. Linear-map based models, additive models and multiplicative models are used. Composite vectors are generated for a group of adjective-noun expressions from the targeted corpus. This group includes either semantically tolerable adjective-noun expressions or not. Additive and multiplicative models gave extraordinary results compared to other models.

Reddy et al. [53], investigated the impact of polysemy in word space models (WSM) for compositionality detection on English language. They presented an exemplar-based WSM that each word is presented by all its corpus samples (exemplars).

Chapter 4

Experimental Results

In this thesis, word vector representations (in other words word embeddings) are used to detect MWEs assuming that vector representations hold the required information on the meaning/sense of the words. In order to examine this, a set of example words/unigrams in surface form of corpus Bilkent is selected randomly. This set involves the words ” abd, kemal, lira, bakanlığı, sistem ”. For each word in this set, its cosine similarity is calculated with all the remaining words in corpus. Then top most 10 words which have the highest similarity scores are selected. In Table 4.1 and Table 4.2, list of 10 words for each type of vector representation are given for surface form and stemmed form of Bilkent corpus respectively. When the words that hold the highest similarity with the given sample words are examined, it is seen that representations may be used to present the words since they are successful in grouping the words that have similar meanings or the words which imply each other due to being used commonly in same concepts.

METHOD	**abd**	**kemal**	**lira**	**bakanlığı**	**sistemi**
CBOW	kesiminin	ibrahim	milyon	dışişleri	yasa
	savunma	salih	milyar	yardımcısı	doğrudan
	konseyi	tayyip	bin	abdüllatif	ekonomi
	birliği	özdemir	trilyon	milli	hak
	bakanlığının	m	dolarlık	kurulu	stand
	yönetim	bakanlar	liraya	milletvekili	birer
	özel	ömer	liralık	başkan	alınması

Continued on next page

Table 4.1 – continued from previous page

METHOD	abd	kemal	lira	bakanlığı	sistemi
	yapan	başkanvekili	dolara	lideri	görüş
	sağlık	yazıcıoğlu	toplam	müdürü	kriz
	büyükelçisi	eşi	yüzde	bakanı	üzerindeki
SG	dışişleri	özdemir	milyar	bakanlığının	esas
	hollanda	erdoğan	milyon	bakanlığına	sorumluluk
	ırak	melih	dolara	maliye	köklü
	birliğinin	tayyip	trilyon	dışişleri	barışı
	bm	hüseyin	dolarlık	gülün	dikkate
	büyükelçisi	eşref	liralık	başkanlığının	yolu
	anlaşması	necdet	liraya	bakanlığında	sağlamak
	askeri	naci	doları	abdüllatif	batılı
	kıbrıstan	çetin	bin	sorumlu	enflasyona
	gümrük	erdoğanın	toplam	sağlık	rejimin
GLOVE	dışişleri	yazıcıoğlu	kazanmıştı	dışişleri	seçim
	makamlarına	müdürü	lirayı	maliye	din
	bakanlığı	tahir	milyon	müsteşarı	akdeniz
	japonya	dsi	dolayında	sözcüsü	içi
	hazinesinin	toyotasa	milyar	bakanları	cinsellik
	ingiltere	aycan	275	bakanlığının	müdürlüğü
	dışişlerinin	alaaddin	liraya	japonya	verenler
	hazine	altınbilek	200	yevgeni	anayasal
	başta	duyar	850	içişleri	istihbarat
	almanya	yazıcıoğlunun	trilyon	bakanlarının	toyotasa
LDA	defa	vakfı	milyon	ölüm	görünen
	nereden	baskı	dış	icat	alarak
	toplantısında	yara	maaş	dedikleri	olacağı
	demek	göze	eli	kanıt	türlü
	mektup	başından	düşmüş	hastanesinde	tbmm
	ayakta	bm	ışık	geçirilmesi	çıktım
	geçiriyor	akıl	adamları	yapılarak	rekabet
	uzlaşma	sol	neler	iflas	yıllardan
	erbakanla	derin	birer	raporlar	pilot
	korkunç	konutunda	olmuş	korkunç	parlamenterler
LSA	kesiminin	berna	milyon	dışişleri	not
	savunma	albay	liraya	attığı	olacaktı
	bm	naci	dolara	sağlık	yeni
	alalım	hayri	milyar	sürpriz	hale
	rum	yazıcıoğlu	liralık	savunma	yasa
	konseyi	muzaffer	dolarlık	vali	olgu

Continued on next page

Table 4.1 – continued from previous page

METHOD	abd	kemal	lira	bakanlığı	sistemi
	rauf	ahmet	bin	bakanlığının	güç
	gitmiyor	genelkurmay	asgari	kırk	temel
	havada	bakanlığında	aylık	büyükelçisi	başkanının
	tipi	yardımcısı	maaş	sakarya	parlamenterler

Table 4.1: Top most 10 words in cosine similarity list of surface form

In Table 4.2 results are demonstrated for stem form of words. It is examined from Table 4.2 that there exists some errors in stemming. We ignore such cases where stemming or other preprocessing tasks bring into the datasets.

METHOD	abd	kemal	lira	bakanlık	siste
CBOW	ırak	özdemir	milyon	mill	batı
	asker	güngör	dolar	bakan	temel
	savunm	hüsey	dolarlık	dışiş	kes
	kuzey	ibrahim	milyar	kuru	kökl
	konse	doç	bin	mali	ekonomi
	ab	yıldır	trilyon	cumhurbaşkan	sorumluluk
	bm	m	liralık	üye	rej
	sanayi	saim	par	sorum	güven
	dışiş	nusret	mark	başkanlık	güç
	büyükelçi	ayhan	faiz	çelik	ortak
SG	temas	güngör	milyar	bakan	kökl
	dışiş	sönmez	dolar	dışiş	uy
	büyükelçi	erdoğan	milyon	abdüllatif	düzen
	çin	muammer	liralık	genelkurmay	düzenle
	hollan	demirç	mark	ulaştır	toplumsal
	rauf	kaymaka	trilyon	alınarak	uygun
	denktaş	burhan	dolarlık	mali	ideolojik
	ırak	elazığ	miktar	usul	temel
	bm	necdet	ödenecek	savunm	sağlamak
	ab	özdemir	cari	açıldık	ilke
GLOVE	dışiş	tahir	milyon	dışiş	uygulan
	ingilter	yazıcıok	milyar	içiş	mevcut

Continued on next page

Table 4.2 – continued from previous page

METHOD	abd	kemal	lira	bakanlık	siste
	iran	aycan	do	bağlanma	siyasal
	japonya	müdür	satılan	müsteşar	demokratik
	pakistan	güvener	trilyon	mali	sınav
	1980lerde	mustaf	200	bakan	laik
	türki	alaaddi	kazanmış	ağrı	temel
	fran	as	ödedik	pangalos	getirilecek
	hazin	bakacak	100	onur	parti
	bakanlık	bilecik	dolay	klaus	eğit
LDA	kaş	akıl	milyon	mazhar	büyükelçilik
	çukurov	atarak	veremedik	habers	edilir
	bozt	olas	trakya	ramazan	roman
	özkök	sess	satt	şık	harç
	nusret	oturan	defter	yardımlaş	inanılmaz
	salim	atmak	yapılarak	londr	sefer
	edemeyecek	bağç	harcayacak	ihtimal	talip
	zamanki	kurtul	milyar	ayr	kaçır
	eşref	buyur	eden	dayanış	havaalan
	seslen	masa	kadir	dış	tekrar
LSA	savunm	hayri	milyon	mali	yönelik
	kesim	berna	milyar	sorum	lise
	bm	albay	dolar	sağlık	bell
	denktaş	dair	liralık	vali	başlangıç
	rum	muzaffer	aylık	mill	atmak
	asker	yazıcıok	bin	jandar	haya
	kuzey	yardımcı	asgari	şener	not
	kıbrıs	nusret	trilyon	yönelik	kes
	rauf	başkan	kret	şube	deney
	taraf	genel	dolarlık	teşhis	yeter

Table 4.2: Top most 10 words in cosine similarity list of stemmed form

	Bilkent-stemmed					combined-stemmed		
	CBOW	SG	Glove	LDA	LSA	CBOW	SG	Glove
Naive Bayes	0,804	0,787	0,810	0,723	0,817	0,750	0,738	0,774
BayesNet	0,815	0,769	0,824	0,786	0,836	0,760	0,749	0,769
SMO	**0,853**	**0,839**	0,831	0,821	0,803	0,811	**0,807**	0,810
Logistic Reg.	0,823	0,808	0,815	0,815	0,841	0,807	0,801	0,800
Voted Perceptron	0,806	0,808	0,816	***0,546***	***0,697***	0,793	0,790	0,795
IBk	0,837	0,822	0,821	0,812	0,773	**0,830**	0,790	**0,826**
AdaBoost	0,807	0,750	0,788	0,822	0,821	0,718	0,717	0,737
OneR	***0,759***	***0,666***	***0,745***	0,816	0,810	***0,616***	***0,615***	***0,661***
J48	0,779	0,722	0,758	0,837	0,808	0,702	0,662	0,710
RandomForest	0,844	0,823	**0,848**	**0,852**	**0,852**	0,819	0,792	0,806
Average	0,813	0,779	0,806	0,783	0,806	0,761	0,746	0,769
Std. Dev.	0,029	0,054	0,032	0,090	0,044	0,067	0,065	0,052

Table 4.3: *F1 scores in stemmed form of corpora*

In this thesis, Bilkent and combined corpora are investigated by using 5 different word embedding methods such as Word2Vec (CBoW, SG), GloVe, LSA, LDA on surface and stem forms of corpora. We tried to answer following questions in our experiments.
1. What will be the performance of classification methods when word vector representations are given as inputs to the classifier?
2. Which vector representations will succeed in detecting MWEs?
3. Which classification method will perform better in detecting MWEs when word vector representations are employed?
4. Does stemming change the performance of MWE detection?

In this chapter, the experimental results are presented via tables holding commonly averaged F1 scores of 5-fold cross validation in order to answer above listed questions.

Table 4.3 involves F1 scores of 10 classification methods in stemmed form of Bilkent and combined corpus. The highest F1 score for each embedding method (each column) is given bold. For example, in stemmed Bilkent corpus when CBOW embeddings is used, the highest performance F1=0.853 is observed by SMO method. Examining the highest F1 scores, it is seen that there exists no classification method that outperforms continuously for all types of embeddings. In order to identify the best performing embedding method in stemmed corpus, the average performance of classification methods and regarding standard deviation values are calculated. These values are given in last two rows in Table 4.3. In

Table 4.3, the underlined values that are given in average row refer to the highest average F1 scores. Considering these highest average F1 values, it can be stated that the best performing embedding method is CBOW and Glove respectively for stemmed Bilkent and combined corpora.

On the other hand, the lowest F1 score for each embedding method is given with italic. For instance, in stemmed Bilkent corpus when LDA embedding is used, the lowest performance F1=0.546 is obtained by Voted Perceptron method. Investigating the lowest F1 scores, it is seen that Voted Perceptron and OneR are not efficient classification methods for all word embeddings on Bilkent and combined corpora. Also it is clearly seen that Voted Perceptron has the lowest F1 scores on LDA and LSA while OneR has the lowest F1 scores for other word embeddings.

	Bilkent-surface					combined-surface		
	CBOW	SG	Glove	LDA	LSA	CBOW	SG	Glove
Naive Bayes	0,781	0,818	0,834	0,698	0,721	0,730	0,793	0,636
BayesNet	0,816	0,808	0,835	0,792	0,861	0,781	0,782	0,660
SMO	0,847	**0,863**	**0,863**	0,825	0,841	**0,847**	0,815	**0,678**
Logistic Reg.	0,802	0,810	0,814	0,810	0,838	0,835	0,810	0,677
Voted Perceptron	***0,771***	0,811	0,826	***0,612***	***0,681***	0,819	0,808	0,673
IBk	0,816	0,830	0,843	0,818	0,822	0,844	**0,816**	0,623
AdaBoost	0,804	0,778	0,825	0,815	0,842	0,733	0,715	0,618
OneR	0,773	***0,654***	***0,766***	0,794	0,831	***0,607***	***0,569***	***0,537***
J48	0,796	0,777	0,802	0,833	0,814	0,701	0,685	0,602
RandomForest	**0,852**	0,839	0,862	**0,866**	**0,869**	0,842	0,811	0,660
Average	0,806	0,799	0,827	0,786	0,812	0,774	0,760	0,636
Std. Dev.	0,028	0,057	0,029	0,075	0,061	0,080	0,081	0,044

Table 4.4: *F1 scores in surface form of corpora*

Table 4.4 involves F1 scores of 10 classification methods in surface form of Bilkent and combined corpus. The highest F1 score for each embedding method (each column) is given with bold. For example, in surface form of Bilkent corpus when CBOW embedding is used, the highest performance F1=0.852 is observed with RandomForest method. Examining the highest F1 scores, though SMO and random forest methods are commonly more successful in classification, still it is not possible to state that there exists one classification method that outperforms continuously for all types of embeddings.

In order to identify the best performing embedding method in surface form of corpora, the average performance of classification methods and regarding standard deviation values are calculated. These values are given in last two rows in Table 4.4. In Table 4.4, the underlined values that are given in average row refer to the highest average F1 scores. Considering these highest average F1 values, it can be stated that the best performing embedding method is Glove and CBOW respectively for Bilkent and combined corpora.

On the other hand, the lowest F1 score for each embedding method is given with italic. For instance, in surface Combined corpus when SG embedding is used, the lowest performance F1=0.569 is obtained by OneR method. Investigating the lowest F1 scores, it is seen that Voted Perceptron and OneR are not efficient classification methods for all word embeddings on Bilkent and combined corpora like in stemmed form of corpora. However, on surface form of Bilkent corpus the

	Bilkent corpus	Combined corpus
CBOW	60%	30%
SG	10%	20%
GLOVE	10%	100%
LDA	50%	-
LSA	30%	-

Table 4.5: *Comparison of classification performances in stemmed and surface form of corpus*

lowest F1 score is obtained by Voted Perceptron when CBOW is used.

Table 4.5 gives the percentage of classification methods where the observed F1 values are higher in stemmed corpus. For example, 60 percent of 10 classification methods generated higher F1 values in stemmed Bilkent corpus compared to the F1 values in surface form of Bilkent corpus. As the values in table is examined, it is seen that especially for SG embeddings, stemming is not required.

Chapter 5

Conclusion

In multiword expressions, words may combine changing/loosing their own meanings to create a new one. In the latest studies on natural language processing, the senses/meanings of the words/word combinations are presented by word vector representations/word embeddings. Though there exists many methods to build vector representations, main idea is similar. While presenting the word/word combinations with vectors, it is assumed that the neighbouring words convey the information concerning the given target word/word combination in language.

In this thesis, we accept MWE detection as a binary classification task and give MWE candidate vector as input to classifiers. The term MWE in this thesis is limited to sequential two-words (bigrams). We construct positive and negative MWE samples (word combinations) list using IMST [[6], [7], [8]] corpus and build up vectors of MWE samples both in surface and stemmed forms using Bilkent [9], Leipzig [10] and Wikipedia. Five different word embedding methods are employed to build vectors. The resulting vectors are used to construct the final vector of MWE candidate and this vector is given as input to ten different classifiers. The classification performance is evaluated by well-known F1 measure. The experimental results revealed that

- Stemming does not improve performance in MWE extraction.

- Considering on average evaluation values GloVe and CBOW representations succeed in experiments.

Since there exists no other studies employing the same corpus (IMST) to build up MWE data set, it is not possible to perform an exact comparison on the evaluation results. However, when our results are compared to the previous MWE extraction studies on Turkish (e.g [49], [48], [50]), it may be stated that word vector representations succeed in detecting MWEs in Turkish. As a further work, we plan running experiments on the same data set utilizing previously proposed methods.

BIBLIOGRAPHY

[1] Manning CD, Schütze H. *Foundations of statistical natural language processing*. MIT Press. England 1999

[2] Firth JR. Modes of meaning. Papers in linguistics 1934-51. Oxford University Press 1967

[3] Sinclair JM. Corpus, concordance, collocation. Oxford University Press. Oxford 1991

[4] Hoey M. *Patterns of lexis in text*. Oxford University Press 1991

[5] Kumova Metin S, Karaoğlan B. Collocation extraction in turkish texts using statistical method. In Advances in natural language processing. Springer; 2010. p. 238-49.

[6] Eryiğit G, İlbay T, Can OA. Multiword expressions in statistical dependency parsing. Proceedings of the Workshop on Statistical Parsing of Morphologically-Rich Languages SPRML at IWPT. 2011 Oct; Dublin. p. 44-55.

[7] Oflazer K, Say B, Hakkani Tur D, Tur G. Building a turkish treebank, Building and exploiting syntactically-annotated corpora, Anne Abeille Editor. Kluwer Academic Publishers; 2003.

[8] Atalay NB, Oflazer K, Say B. The annotation process in the turkish treebank. Proceedings of the EACL Workshop on Linguistically Interpreted Corpora(LINC); Apr 13-14; Budapest; Hungary 2003

[9] Tür G, Hakkani-Tür D, Oflazer, K. A statistical information extraction system for turkish. Nat Lang Eng. 2003; 9(2): 181-210

[10] Quasthoff U, Richter M, Biemann C. Corpus portal for search in monolingual corpora. Proceedings of the 5th Int. Conf. on Lang. Resources and Evaluation 2006 May; Genoa, Italy.

[11] Pennington J, Socher R, Manning C. GloVe: global vectors for word representation. Proceedings Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. p. 1532-43.

[12] Web site link: https://github.com/otuncelli/turkish-stemmer-python, Last Access [2018 Nov 11]

[13] Frank E, Hall MA, Witten IA. Data mining: Practical Machine Learning Tools and Techniques. Fourth Edition. Morgan Kaufmann Publishers; 2016

[14] John GH, Langley P. Estimating continuous distributions in bayesian classifiers. Proceedings Eleventh Conference on Uncertainty in Artificial Intelligence; 1995; San Mateo. p. 338-45.

[15] Pearl J. Bayesian Networks, UCLA Cognitive Systems Laboratory. Technical Report In MIT Encyclopedia of the Cognitive Sciences. Cambridge. MA. 1999

[16] Friedman N, Geiger D, Goldszmidt M. Machine learning. 1997 Nov; 29(2-3):131-163 doi:10.1023/A:1007465528199

[17] Dolgun MÖ, Ersel D. Doğrudan pazarlama stratejilerinin belirlenmesinde veri madenciliği yöntemlerinin kullanımı. İstatistikçiler dergisi: istatistik ve aktüerya 2014; 7: 1-13

[18] Bouckaert RR. Weka reports/manual: bayesian network classifiers in weka for version 3-5-7. University of Waikato; 2008. Available from: http://www.cs.waikato.ac.nz/ remco/WEKA.bn.pdf

[19] Platt JC. Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf B, Burges CJC, and Smola A, editors. Advances in kernel methods – support vector learning. MIT Press; 1998

[20] Shevade SK, Keerthi SS, Bhattacharyya C, Murthy KRK. Improvements to the SMO algorithm for SVM regression. IEEE Transactions on Neural Networks. 2000; 11(5):1188-93

[21] Smola AJ, Schoelkopf B. A tutorial on support vector regression. 1998

[22] Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V. Support vector regression machines. Proceedings of the 9th International Conference on Neural Information Processing Systems. 1996 Dec 3-5;Denver, Colorado. p.155-61.

[23] Tabachnick BG, Fidell LS. Using multivariate statistics. Fifth Edition. Boston: Pearson Education, Inc. 2007

[24] Le Cessie S, van Houwelingen, JC. Ridge estimators in logistic regression. J Appl Stat. 1992; 41(1): 191-201.

[25] Helmbold DP, Warmuth MK. On weak learning. J Comput Syst Sci. 1995; 50(3): 551–73

[26] Freund Y, Schapire RE. Large margin classification using the perceptron algorithm. In 11th Annual Conference on Computational Learning Theory. 1998; New York. NY. p. 209-217.

[27] Freund Y, Schapire RE. Experiments with a new boosting algorithm. In Thirteenth International Conference on Machine Learning. San Francisco; 1996 p. 148-56.

[28] Schapire RE. Explaining AdaBoost. In Schölkopf B, Luo Z, Vovk V, editors, Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik. Springer; 2013

[29] Eible G, Pfeiffer KP. Multiclass boosting for weak classifiers. J Mach Learn Res. 2005; 6:189–210.

[30] Iba W, Langley Pat. Induction of one-level decision trees. In ML92: Proceedings of the Ninth International Conference on Machine Learning, 1992 Jul 1–3; Aberdeen, Scotland. San Francisco, CA: Morgan Kaufmann; p. 233–40.

[31] Holte RC. Very simple classification rules perform well on most commonly used datasets. Machine Learning. 1993; 11: 63-91.

[32] Quinlan R. C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers. San Mateo. CA. 1993

[33] Archer KJ. Emprical characterization of random forest variable importance measure. Comput Stat Data Anal. 2008; 52(4): 2249-60.

[34] Breiman L. “Random Forests”. Machine Learning. 2001; 45(1):5-32

[35] Breiman, L, Cutler A. Random forest. 2005

[36] Oflazer K, Çetinoğlu Ö, Say B. Integrating morphology with multi-word expression processing in turkish. Proceedings of the Workshop on Multiword Expressions: Integrating Processing. 2004. p. 64-71.

[37] Tsvetkov Y, Wintner S: Identification of multi-word expressions by combining multiple linguistic information sources. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011 Jul 27-31; Edinburg, Scotland, UK. p. 836-45.

[38] Sarıkaş F. Problems in translating collocations. Elektronik Sosyal Bilimler Dergisi www.e-sosder.com 2006; 5(17): 33-40.

[39] Kumova Metin S. Feature selection in multiword expression recognition. Expert Syst Appl. 2018; 92: 106-23.

[40] Pecina P. A machine learning aproach to multiword expression extraction. Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions. 2008

[41] Bouma G. Collocation extraction beyond the independence assumption Proc ACL 2010 Conf Short Pap. 2010; p.109-14.

[42] Cordeiro S, Ramisch C, Villavicencio A. Mwetoolkit+sem: integrating word embeddings in the mwetoolkit for semantic MWE processing. Language Resources and Evaluation (LREC): 2016 May; Portorož, Slovenia.

[43] Ramisch C, Villavicencio A, Boitet C. Mwetoolkit: a framework for multiword expression identification. Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10): 2010 May; Valletta, Malta.

[44] Kumova Metin S. Enlarging multiword expression dataset by co-training. Turk J Elec Eng and Comp Sci. 2018; 26(5): 2583-94.

[45] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. Proceedings 11th Annual Conference on Computational Learning Theory: 1998; Madison, Wisconsin, USA. p. 92-100

[46] Kumova Metin S. Standard co-training in multiword expression detection. Proceedings International Conference on Intelligent Human Computer Interaction; 2017 Evry, France. p. 178-88.

[47] Kumova Metin S, Karaoğlan B. Identifying collocations in turkish using statistical methods. Bilig - Turk Dünyası Sosyal Bilimler Dergisi. 2016; 78(78): 253-286.

[48] Uymaz Aka H, Kumova Metin S. A comprehensive analysis of web-based frequency in multiword expression detection. IJISAE, 2017; 5(3):145-53.

[49] Kumova Metin S. Neighbour unpredictability measure in multiword expression extraction. Comput Syst Sci and Eng. 2016; 3: 209-21.

[50] Eren LT, Kumova Metin S. Vector space models in detection of semantically non-compositional word combinations in turkish. Proceedings of 7th International Conference - Analysis Of Images, Social Networks And Texts; 2018. p. 53-63.

[51] Pedersen T. Identifying collocations to measure compositionality. 1994; 13(1):11-24.

[52] Vecchi EM, Baroni M, Zamparelli R. (Linear) Maps of the impossible: capturing semantic anomalies in distributional space. Proceedings DiSCo '11 Proceedings of the Workshop on Distributional Semantics and Compositionality; 2011. p. 1-9.

[53] Reddy S, McCarthy D, Manandhar S, Gella S. Exemplar-based word-space model for compositionality detection: shared task system description, Proceedings of the Workshop on Distributional Semantics and Compositionality (DiSCo '11); 2011 Jun 24; Portland, Oregon. p. 54-60.

[54] Web site link: https://israelg99.github.io/2017-03-23-Word2Vec-Explained/, Last Access [2019 Sep 15]

[55] Web site link: https://medium.com/@mubuyuk51/word2vec-nedir-türkçe-f0cfab20d3ae, Last Access [2019 Sep 15]