T.C.
MELIKSAH UNIVERSITY

THE GRADUATE INSTITUTE OF SCIENCES AND ENGINEERING

M.Sc. THESIS IN ELECTRICAL AND COMPUTER ENGINEERING

TWEET AND USER RECOMMENDATION IN TWITTER

Fahriye GEMCİ FURAT

Master's Thesis

June 2015
KAYSERİ

TWEET AND USER RECOMMENDATION IN TWITTER

by

Fahriye GEMCİ FURAT

A thesis/dissertation submitted to

the Graduate Institute of Sciences and Engineering

of

Meliksah University

in partial fulfillment of the requirements for the degree of

Master of Science of Engineering

in

Electronics and Computer Engineering

June 2015
Kayseri, Turkey

APPROVAL

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

______(Signature)_____
(Title and Name)
Head of Department

This is to certify that I have read this thesis/dissertation and that in my opinion it is fully adequate, in scope and quality, as a thesis/dissertation for the degree of Master of Science of Engineering.

______(Signature)_____
(Title and Name)
Supervisor

Examining Committee Members

(Title and Name) (Signature)______________________

(Title and Name) (Signature)______________________

(Title and Name) (Signature)______________________

(Title and Name) (Signature)______________________

(Title and Name) (Signature)______________________

It is approved that this thesis/dissertation has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

______(Signature)_____
(Title and Name)
Director

June 2015

TWEET AND USER RECOMMENDATION IN TWITTER

Fahriye GEMCİ FURAT

M.Sc. Thesis Electronics and Computer Engineering
June 2015

Supervisor: Asst. Prof. Dr. Kadir Aşkın PEKER

Co-Supervisor:

ABSTRACT

In the information age, we can retrieve information fairly easily through the network. We are concerned with information in Social Networks in this thesis. Twitter is one of the latest trends of social media in the globalized world. Since 2006, with over 500 million users as of 2012 and 340 million tweets daily, Twitter is a great information source for researchers. Hence we prefer Twitter from social media.

Twitter users who are interested in same topics follow each other. On the other hand, sometimes the users are interested in some different topics from each other. In this case, the users find information that they don’t want. In order to solve this problem, we design a framework that recommends tweets and users to other users by using similarity of the users’ tweets.

Recently, Latent Dirichlet Allocation (LDA) has been successfully used in analysis of tweet topics in English. However, LDA hasn’t been applied to Turkish tweets. In this thesis, LDA analysis is used for Turkish tweets to implement a recommender system for the users in Turkish language.

Turkish is an agglutinative language, which makes application of LDA a new challenge compared to LDA on English tweets. Hence, a series of preproccessing steps were performed to make the tweet texts suitable for LDA analysis. We show promising results in this thesis.

Keyword: Social Media, Twitter, Recommender System, Content Based Filtering, Latent Dirichlet Allocation, k-means

TWITTER'DA TWEET VE KULLANICI ÖNERİ SİSTEMİ

Fahriye GEMCİ FURAT

Yüksek Lisans Tezi Elektronik ve Bilgisayar Mühendisliği
Mayıs 2015

Tez Yöneticisi: Yrd. Doç. Dr. Kadir Aşkın PEKER

Ortak Tez Yöneticisi:

ÖZ

Bilgi çağında, ağ üzerinden bilgiyi çok kolay bir şekilde elde edebiliriz. Bu tezde sosyal ağdaki bilgi üzerinde çalıştık. Küreselleşen dünyada, Twitter sosyal medyanın son trendlerinden biridir. Twitter 2006'dan itibaren, 2012 verilerine dayanarak 500 milyonun üzerinde kullanıcısıyla ve günlük 340 milyon tweetiyle araştırmacılar için büyük bir bilgi kaynağıdır. Bu yüzden sosyal medyadan Twitter'ı tercih ettik.

Aynı konularla ilgilenen Twitter kullanıcıları birbirini takip eder. Ancak, bazen kullanıcılar birbirlerinden farklı konularla da ilgilenirler. Bu durumda, kullanıcılar istemedikleri bilgiye erişirler. Bu problem çözmek amacıyla, kullanıcıların tweetlerin benzerliğinden yola çıkarak tweetleri ve kullanıcıları diğer kullanıcılara tavsiye eden bir yapı tasarladık.

Son zamanlarda, Latent Dirichlet Allocation (LDA) İngilizce tweet konu analizinde başarılı bir şekilde kullanılmıştır. Ancak LDA Türkçe tweetler için uygulanmamıştır. Türkçe kullanan kullanıcılara yönelik bir öneri sistemi oluşturmak amacıyla bu tez çalışmasında Türkçe tweetler için LDA analizi kullanılmıştır.

İngilizce tweetlerde LDA kullanılması, Trükçe tweetlerde LDA kullanılması ile karşılaştırıldığında, Türkçenin sondan eklemeli bir dil olması nedeniyle, İngilizce için başarılı olan LDA için yeni bir uygulama yapılmıştır. LDA analizi için tweet metinlerini uygun hale getirmek amacıyla bir seri önişlem adımları kullandık. Bu tez çalışmasında elde edilen umut verici sonuçlar sunulmaktadır.

Anahtar Kelimeler: Sosyal Medya, Twitter, Tavsiye Sistemi, İçerik Tabanlı Filtreleme, Latent Dirichlet Allocation, k-means

ACKNOWLEDGEMENT

I am grateful to Kadir A. PEKER, who is my advisor in the department of Computer Engineering. I would like to thank him for sharing expertise, valuable guidance and engagement through the learning process of this master thesis. I would like to thank Ahmet UYAR since I have learned many things during my master's student. I would like thank Oğuzhan AKGÖL for support.

I would like to thank Murat FURAT, my husband for the patience. Thank you for the unconditional love and support. I would like to thank Gül GEMCİ and Alaettin GEMCİ my parents and Aykut Mahmut GEMCİ my brother for encouragement throughout my life. I would like to thank my teachers and lecturers, my friends for their valuable support. I cannot finish master program without you. I would like to thank you all for you are with me.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION	
LDA	Latent Dirichlet Allocation
API	Application Program Interface
NLP	Natural Language Processing
BoW	Bag-of-Words
LSA	Latent Semantic Analysis
SVD	Singular Value Decomposition
MCMC	Markov Chain Monte Carlo
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Social media have become an indispensable part of life at the present time since millions of people communicate through social media. Twitter is one of the most popular and active social networking sites. Facebook, Wikipedia, YouTube and last fm are other similar platforms.

Twitter is a type of micro-blogging website created by Jack Dorsey in March 2006 where users communicate and interact online by sending and reading at most 140 characters messages called “tweets”. Twitter users share their feelings, opinions and what they do with other users by sending tweets. In other words, Twitter users tell their life to others using tweets. Today, more than 500 million users post nearly 340 million tweets daily [1]. Researchers have performed lots of studies through Twitter since it is a very large and useful resource for text topic analysis and text similarity. Also, Twitter provides an API to access some of its data. Because of this reason, Twitter is preferred as the source of content in this thesis.

Another reason that we target tweets is that it is hard to retrieve information that we want in Twitter. In fact, trend topics and followers can be used to topic analysis and text similarity. However, it is not enough when considering existence of variety of information. Twitter users may have different interests. Therefore, the user follows the other users who are interested in the same topics. But the other users may have not only the same interests but also different interests. Hence, they publish tweets about different topics. Consequently, a user can receive irrelevant messages with the other users having not only same interest but also different ones. Unfortunately, users have reached tweets which they don’t interest.

Spam tweets are another reason we consider tweet topic analysis and recommendation. Spammers publish the spam tweets for different aims. One of them is sending tweets for advertisement. Sometimes users share their messages with different intentions such as pointless babble, jokes, chat, and news. In this sense, the objective of the present thesis rises on the requirement of topic analysis and text similarity.

1.2 OBJECTIVE

In this thesis, our objective is to develop a recommender system for Turkish Twitter users. Although Twitter provides hashtags, trend topics, and follower lists in order to find relevant tweets and users, we think that these are not enough to find similar or wanted tweets and users. The users who are interested in same topic can be interested in different topics at the same time. In this case, when the users follow each other, they get irrelevant topics, too. To prevent this, the recommender system can also serve as a filter.

The recommender system will be based on similarity. The recommender system will find similar tweets and similar users. A Twitter user can be recommended another similar user; or if a user is interested in a tweet, then other similar tweets can be recommended. Similar tweets mean that tweets are relevant; and similar users means that the tweets of those users are relevant. Thus, tweet-tweet similarity and user-user similarity needs to be calculated in order to recommend tweets and users. Similar tweets are usually about the same topic. So our objective then is to find if two tweets are about the same topic, or if two users tweet about the same topics. For this, we should be able to extract the topics of Turkish tweets.

1.3 OUR APPROACH

Twitter users see irrelevant users and tweets as well as relevant ones. Although a user can get only the tweets of the users that he/she prefers, this isn’t enough to get only relevant tweets among all posts of the relevant users. Also, there may be relevant tweets for a user posted by another user that he/she is not following. Hence, a recommender system that selects only relevant tweets among all posts of the relevant users is required for Twitter.

In addition, Twitter has a limited set of options for identifying new people to follow. The users need mechanisms to discover new people if they are relevant users. Due to these reasons, a recommender system for users with Turkish language is developed in this thesis work. The tweet recommendation system finds tweets with similar meaning and on similar topics. At the same time, the user recommendation system finds users who publish tweets with similar topics and recommend these users to each other.

In the new system, a Twitter topic analysis approach is performed. We use Latent Dirichlet Allocation (LDA) for topic detection in tweets. Then we find the similarity of tweets and users based on topic distribution extracted from tweets. This approach is formulated as a classification and clustering application. The difference of the approach from other classification and clustering applications on topic analysis is that this is the first tweet topic analysis work in Turkish that we know of.

The recommender system finds similar tweets and users for Twitter users. The following are the steps of the system in order to perform the recommender system:

- Crawling tweets from Twitter with Twitter API
- Obtaining appropriate root of words by tweet-preprocessing
- Extracting tweet topics using LDA
- Performing classification and clustering of tweets
- Calculating tweet-tweet similarity and user-user similarity

Results can be used to recommend tweets using tweet-tweet similarity and users to other users using user-user similarity.

1.4 CONTRIBUTION

In this thesis, Turkish users and their tweets in Turkish are taken into account. The most important contribution of this thesis is developing a recommender system for Turkish language tweets and users. Similar work has been performed in the previous work for other languages [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. In addition, this study presents how tweets in Turkish language can be processed for topic analysis and similarity computation.

1.5 STRUCTURE OF THE THESIS

In the second chapter, previous studies from the literature are presented. Then, components and algorithms used in the proposed recommender system are given. In the third chapter, the collection of experimental data and design procedure of the proposed system is explained. In the fourth chapter, the results are discussed and future work is outlined.

CHAPTER 2

METHODOLGY AND APPLICATIONS

2.1 RELATED WORK

In recent years, the interest of researchers in social media has increased due to the existing large data of social media that people share. Twitter is one of the popular social media sites that have grown rapidly [13, 14]. Because of this, many researchers have focused on using Twitter on different fields such as education [15], earthquake alert [16], cultural differences detection [17] and stock market prediction [18] as a microblogging platform.

The rapidly growing social media brings a different problem together. The problem is to retrieve information that we are interested in this large data. In other words, the main problem in Twitter is how to search data in the posts and effectively categorize posts. Topic detection is a promising method for determining the similarity and relevance of tweets. Blei et al. [19] developed Latent Dirichlet Allocation (LDA) algorithm for topic modeling of large documents. LDA is used as a standard tool for topic modeling [1]. Bhattarai demonstrates in [1] detects topics of tweets and clusters tweets using LDA. There are many studies in the literature that use LDA in a similar way [20, 21, 22]. In [20], Ramage et.al extract topics of tweets using Labeled LDA [21]. The same authors in [22] map the content of Twitter feeds into dimensions using Labeled LDA.

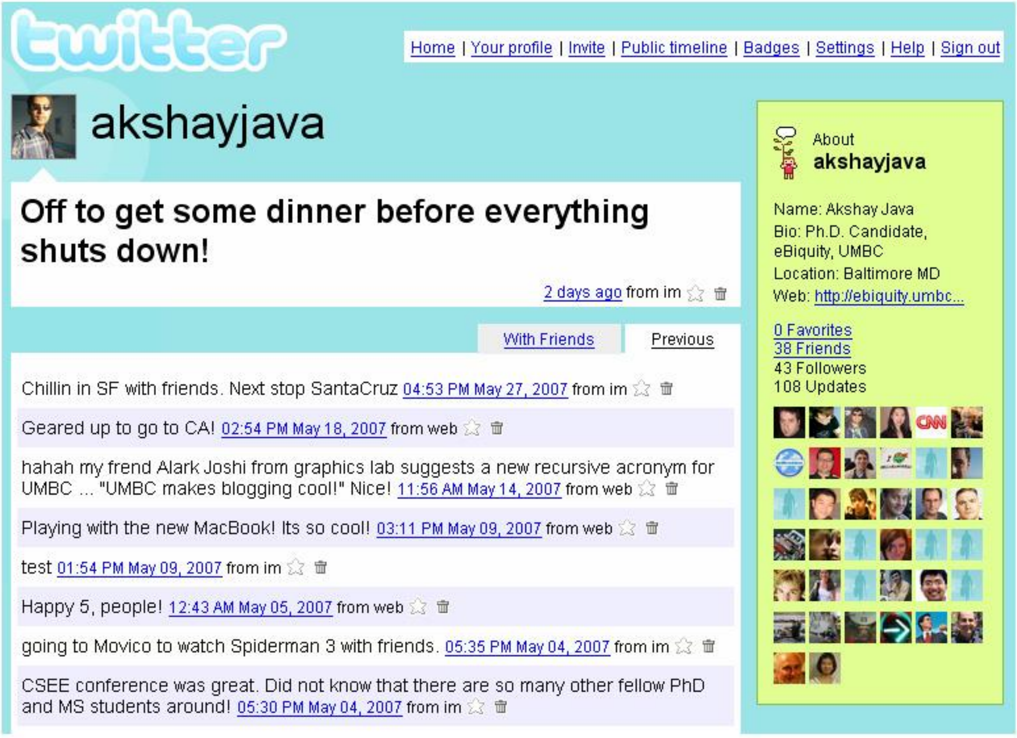
Recommendation Systems in Twitter have been built in previous studies. There have been numerous research work to calculate user similarity and tweet similarity. Some of the user, tweet and hashtag recommendation systems that can be found in the literature are as follows. Pennacchiotti and Siva presented a system that recommends new friends having similar interests to a user [12]. In this system, extracting topics of tweets adapting LDA in [19] was used where documents were preferred instead of users' streams. Zangerle et al. presented a hashtag recommendation system in Twitter [11]. The initial studies showed a success of about 45-50% which can be improved in future work.

2.2 TWITTER

2.2.1 Review of Twitter

Twitter is the most popular microblogging service as a social media [13,23] which was created by Jack Dorsey, Evan Williams, Biz Stone and Noah Glass in March 2006 [24] and incorporated on April 19, 2007 [25].

People use twitter to talk and share about their daily activities [13] as shown in figure 2.1. Twitter users see other people's opinions, aims, interests, and what happens in their lives. This opens Twitter to the use of a wide crowd of people. The data is shared with posts restricted to 140 characters named as a "tweet" . Twitter can be reached using a Web Browser, SMS, e-mail or other applications.


twitter
Home | Your profile | Invite | Public timeline | Badges | Settings | Help | Sign out
With Friends
Previous

akshayjava

Off to get some dinner before everything shuts down!

2 days ago from im

Chillin in SF with friends. Next stop SantaCruz 04:53 PM May 27, 2007 from im

Geared up to go to CA! 02:54 PM May 18, 2007 from web

hahah my frend Alark Joshi from graphics lab suggests a new recursive acronym for UMBC ... "UMBC makes blogging cool!" Nice! 11:56 AM May 14, 2007 from web

Playing with the new MacBook! Its so cool! 03:11 PM May 09, 2007 from web

test 01:54 PM May 09, 2007 from im

Happy 5, people! 12:43 AM May 05, 2007 from web

going to Movico to watch Spiderman 3 with friends. 05:35 PM May 04, 2007 from im

CSEE conference was great. Did not know that there are so many other fellow PhD and MS students around! 05:30 PM May 04, 2007 from im

About
akshayjava

Name: Akshay Java
Bio: Ph.D. Candidate, eBiquity, UMBC
Location: Baltimore MD
Web: http://ebiquity.umbc...

0 Favorites
38 Friends
43 Followers
108 Updates

Figure 2.1 A sample homepage of a Twitter user

2.2.2 Nomenclature

There are lots of components in Twitter. Here we define the following terms: tweet, twitterer, hash-tag, user, follower, following, retweet, URL and trend topic [13,14,26].

A tweet has its own structure. The **tweet** is a prose restricted to 140 characters. Tweets are written by Twitter users. A tweet may also contain a short URL, a hashtag, retweet or a trend topic.

Twitterer is a user who can send and receive messages via the web, SMS, instant messaging clients, and by third party applications.

Hash-tag is a word or phrase that is a topic determined by users.

Hash-tagging is a simple way for users to categorize tweets. It is symbolized by the hashtag character “#”. One tweet may also contain multiple hash-tags followed by the tag itself.

Re-tweeting is to forward a tweet of another user.

Follower is a user who follows another Twitter user.

Following is to follow the other Twitter users.

Trending Topic is the top ten popular terms or topics of discussion at any given moment which Twitter allows users to observe. Trend topic is also a topic that is higher rated than other topics

2.2.3 Twitter API

The Twitter application program interface (API) itself allows the integration of Twitter with other Web services and applications [27].

Twitter provides two APIs for Twitter users. They are REST and Streaming. The REST API provides programmatic access to read and write Twitter data. The Twitter Search API is a type of Twitter's v1.1 REST API [28]. Our data is crawled using SEARCH REST API. The usage of the API is free of charge, it only requires an active Twitter account. Every Twitter account is limited to 20,000 requests per hour. Twitter does not make data older than a week available.

2.2.4 Usage Statistics

Twitter's popularity increases very fast. Overall, it now has 255 million active monthly users, more than nine million users joined into Twitter in the previous quarter (first quarter of 2014) according to Figure 2.2.1 [29].

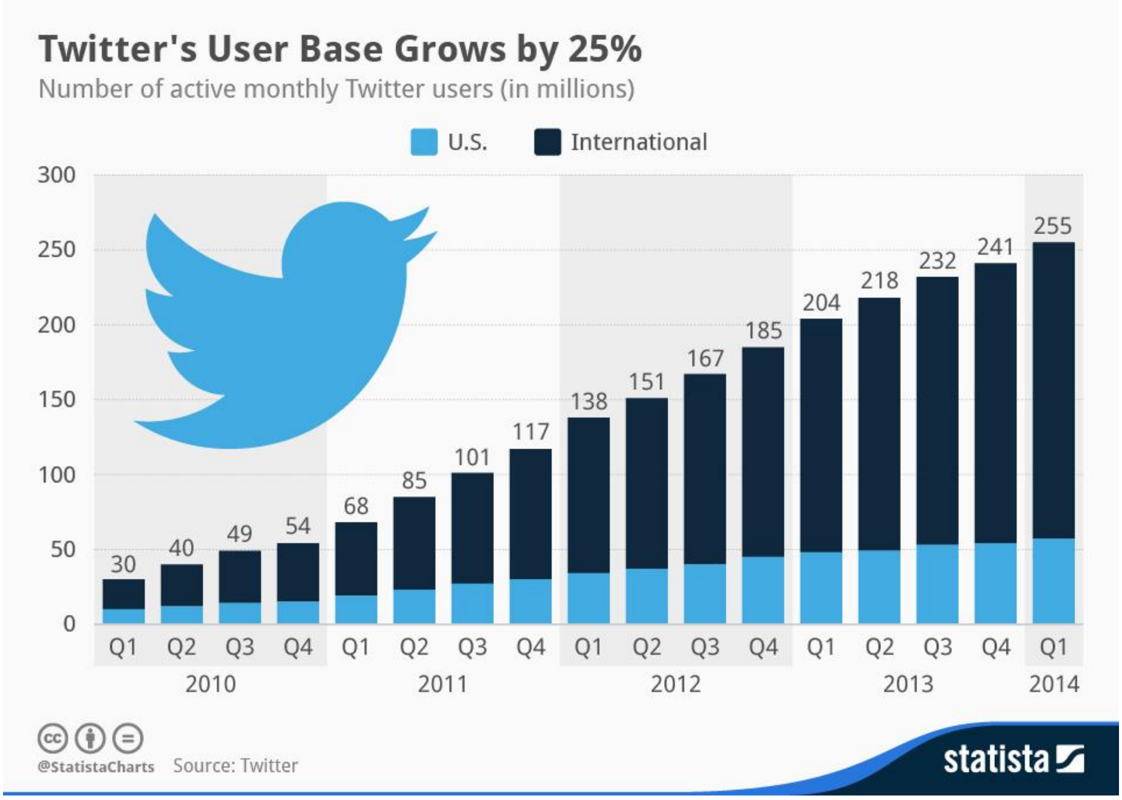


Figure 2.2 Number of active Twitter Users from 2010 to 2014

Number of tweets per day in 2014 was 500 million [25-30]. Number of users according to Twitter is about 500 million in which 288 million of them actively uses Twitter. Figure 2.3 shows the trend of Twitter users from 2010 to 2022. 40% of users worldwide simply use Twitter as a "curated news feed of updates that reflect their passions"[31].

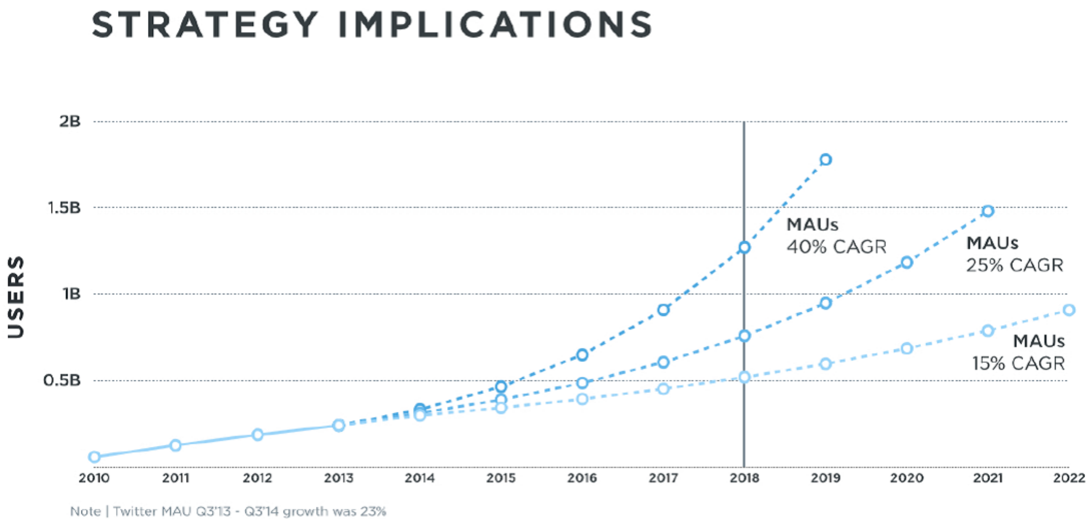


Figure 2.3 Trend of twitter users from 2010 to 2022

2.3 ZEMBEREK

Zemberek is a library designed for Natural Language Processing (NLP) of Turkic languages, especially Turkish having properties of platform independent and open source. The library is used to find root of words in Turkish tweets in this study. The structure of Zemberek software is illustrated in Figure 2.4 [32]. Fundamental NLP operations such as spell checking, morphological parsing, stemming, word construction are provided by Zemberek [32].

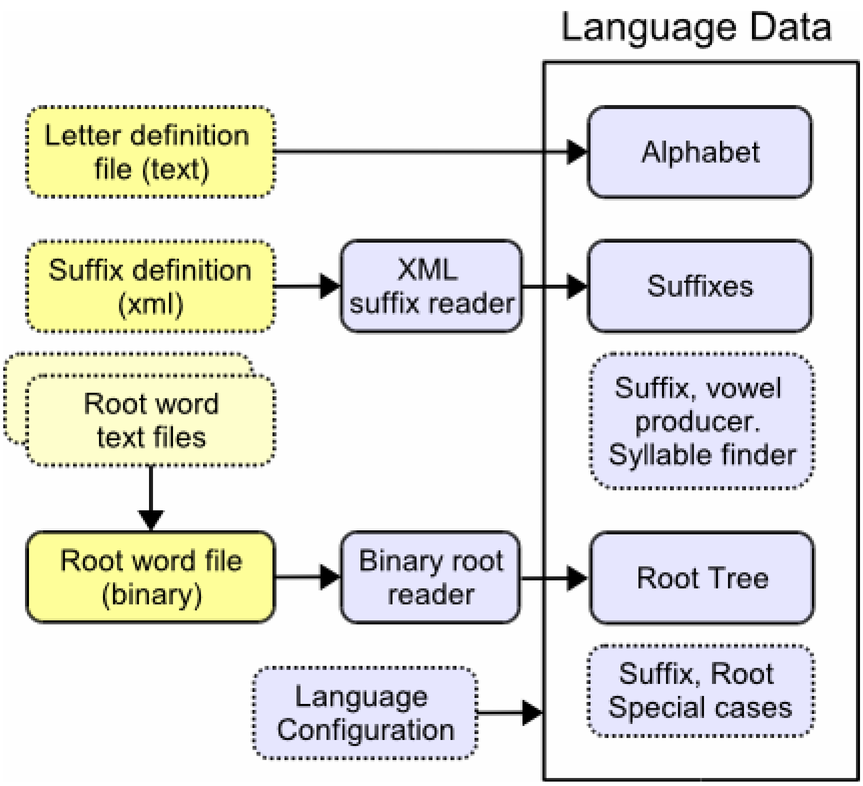


Figure 2.4 Zemberek Program Structure

2.4 TEXT ANALYSIS METHODS

2.4.1 Document representation with Bag-of-Words (BoW) model

The Bag-of-Words (BoW) model is used in data mining, computer vision, information retrieval and natural language processing. BoW model is the most common technique to represent text. In the BoW model a document is considered as consisting of orderless list of words. Each word is represented as a feature. This process is called “Tokenization” [33].

BoW model has a dictionary. The dictionary consists of words of all documents. The document is represented as a row matrix that is the frequencies in the document of words from dictionary.

A group of features are extracted as a feature vector for the document. Since the vector becomes too large, there are several ways to shorten the vector such as stop words removal and stemming [33].

In the following example, an application of BoW model is shown. Assume that there are two sentences as the documents represented in BoW model.

First Sentence: Ali hayvanlardan korkmasına ragmen Ali ata binmek istiyor.

Second Sentence : Ayşe ata kimsenin yardımı olmadan binmek istiyor.

The dictionary consists of words in these sentences as follows.

Our dictionary: (“Ali:2”, ”hayvanlardan:1”, ”korkmasına:1”, ”rağmen:1”, ”ata:2”, “binmek:2”, ”istiyor:2”, ”Ayşe:1”, ”kimsenin:1”, ”yardımı:1”, ”olmadan:1”)

The dictionary has ten words. Therefore, length of the feature vector must be 10. Then BoW model of two documents can be obtained as follows.

Bow model representation of first sentence: [2,1,1,1,1,1,1,0,0,0,0]

Bow model representation of second sentence: [0,0,0,0,1,1,1,1,1,1,1]

After the document has been represented as a BoW model, various classifier or clustering algorithms can be performed.

2.4.2 Similarity measures

The similarity measure gives quantitative proximities or differences between objects. Decision of a similarity measure is also crucial for the desired result. There are lots of different measure models in literature such as Euclidian distance, cosine similarity, Jaccard coefficient, Pearson correlation coefficient and averaged Kullback-Leibler divergence [34].

Similarity functions are commonly used in recommender systems to measure the degree of similarity between two items or users [35]. Cosine similarity is one of the most popular similarity measures for text documents [34].

Cosine similarity is a measure of similarity between two vectors, if documents are represented as term vectors. It measures the cosine of the angle between them. The cosine is between 0 and 1 [34]. If the similarity is 1, two vectors have the same orientation. If the similarity is 0, than the angle between the two vectors is 90° and the documents do not share any words. In other words there are no same words. The other possibility is that if the similarity is -1, two vectors are diametrically opposed. In particular, cosine similarity is used in positive space.

If any two given documents d_j and d_k are represented, as vectors, then their similarity is:

$$sim(d_j, d_k) = \cos(\vec{d}_j, \vec{d}_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} \tag{2.1}$$

Representing the documents with term weights (e.g., tf-idf) as coordinates can be obtained as:

$$sim(d_j, d_k) = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}} \tag{2.2}$$

A query q as a document d_q can be regarded and used in the same formula:

$$sim(d_j, d_q) = \frac{\vec{d}_j \cdot \vec{d}_q}{\left|\vec{d}_j\right|\left|\vec{d}_q\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \tag{2.3}$$

Computation of the cosine similarity example is given below. In Table 2.1, BoW Model of the text example is given.

Table 2.1 BoW Model of the text example

	Ali	hayvanlardan	korkmasına	rağmen	Ata	binmek	istiyor	Ayşe	kimsenin	yardımı	olmadan
X	1	1	1	1	1	1	1	0	0	0	0
Y	0	0	0	0	1	1	1	1	1	1	1

x = (1, 1, 1, 1, 1, 1, 0, 0, 0, 0)
y = (0, 0, 0, 0, 1, 1, 1, 1, 1, 1)

x .y = 1*0 + 1*0 + 1*0 + 1*0 + 1*1+1*1+0*1+0*1 +0*1 +0*1 = 2

||x|| = sqrt(1*1+1*1+1*1+1*1+1*1+1*1+0*0+0*0+0*0+0*0) = 6

||y|| = sqrt(0*0+0*0+0*0+0*0+1*1+1*1+1*1+1*1+1*1+1*1)= 6

cos_sim(x,y)=2/6*6=0,06

2.4.3 Classification using Naive Bayes

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem. The most important assumption in Naive Bayes classifier is that parameters of each conditional distribution are independent. It is accepted that all parameters are equally important. In spite of assumptions that may not hold in reality, classification with Naive Bayes performs better in many complex situations.

In Naive Bayes classifier, each sample is represented by an n-dimensional vector, X = {x_1, x_2, . . ., x_n}.

There are k classes, C_1, C_2, . . . , C_k.

When a sample X given to a Naive Bayes classifier, the classifier will predicts to which class X belongs by maximizing $P(C_j \mid \mathbf{x})$. It is called the maximum posteriori hypothesis.

Bayes' theorem is given as follows:

$$P(C_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_j)P(C_j)}{p(\mathbf{x})} = \frac{p(\mathbf{x} \mid C_j)P(C_j)}{\sum_k p(\mathbf{x} \mid C_k)P(C_k)} \tag{2.4}$$

Only $p(\mathbf{x} \mid C_i)P(C_i)$ is maximized, because of the assumption that the classes are equally likely, that is, $P(C_1) = P(C_2) = . . . = P(C_k)$.

In order to reduce to compute $p(\mathbf{x} \mid C_i)$, class conditional independence is assumed. This means that

$$P(X \mid Ci) \approx \prod_{k=1}^{n} P(xk \mid Ci) \tag{2.5}$$

The probabilities $P(x_1|C_i)$, $P(x_2|C_i)$, . . . , $P(x_n|C_i)$ can be estimated from the training set.

In order to find the class label of X, $P(X|C_i)P(C_i)$ is performed for each class C_i. The highest probability gives suitable class [36, 37].

2.4.4 Clustering with k-means

A simple way to cluster is k-means which is an unsupervised learning algorithm proposed by J.B. MacQueen in 1967 [38, 39]. It is widely used in the literature although it was developed about 50 years ago.

k-means algorithm aims to partition n samples into k clusters, $k<n$ [40]. In other words, each observation belongs to a cluster. Appointment mechanism of k-means gives permission that every data can only belong to a cluster. In k-means clustering,

given a set of *n* data points in d-dimensional space and a positive integer *k* denoting the number of clusters and the problem is to determine a set of *k* points in d-dimensional space, called centers, so as to minimize the mean squared distance from each data point to its nearest center [40].

Figure 2.5 shows how the k-means clustering algorithm works. In k-means algorithm, the initial step is to choose a set of *k* instances as centers of the clusters. The user usually determines the *k* value. Some criteria can be used to automatically estimate *k*. It is an approximation to an NP-hard combinatorial optimization problem. Second step is that the training samples randomly or systematically are assigned. Any initial partition is chosen to classify the data into *k* clusters. Different initial seed sets can result in different final partitions. The following steps are performed for this process.

1. The first k training sample as single-element clusters is taken.
2. Each of the remaining (*N*-*k*) training samples is assigned to the cluster with the nearest centroid. After each assignment, the centroid of the gaining cluster is recomputed.

Third step is to take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, this sample is switched to that cluster and updated the centroid of the cluster gaining the new sample and the cluster losing the sample. The iterative relocation can continue from the new partition until no more relocations occur.

The cluster centroids are recalculated either after each instance assignment or after the whole cycle of re-assignments. This process is iterated.

K-means is naturally an iterative algorithm. It converges only to a local minimum. It works only for numerical data. Implementing k-means is easy [34, 38, 39, 40, 41].

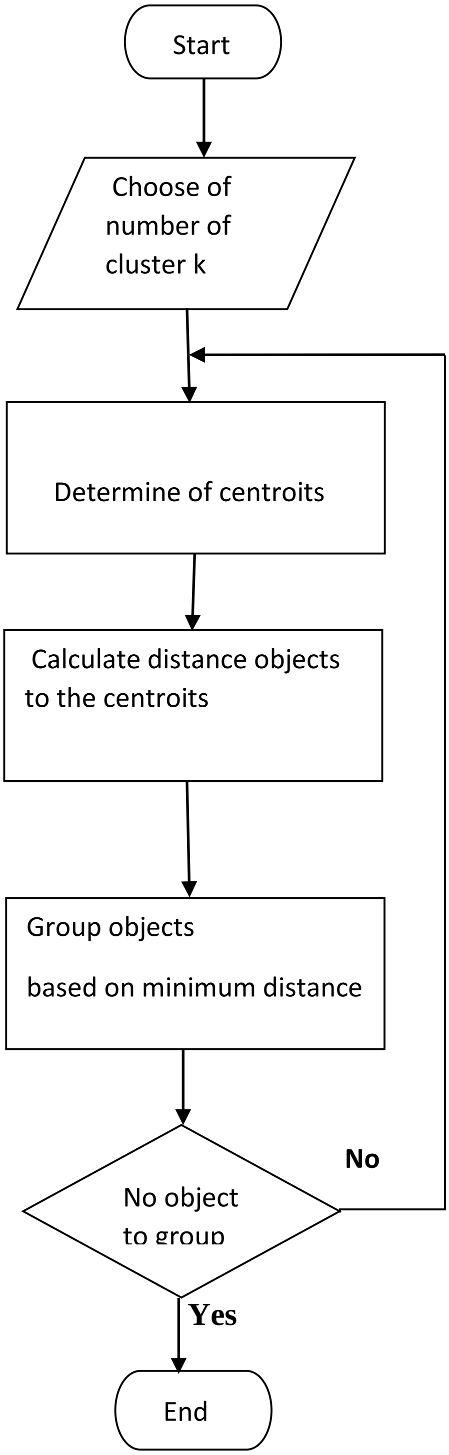

Start
Choose of number of cluster k
Determine of centroits
Calculate distance objects to the centroits
Group objects based on minimum distance
No object to group
No
Yes
End

Figure 2.5 k-means algorithm.

2.4.5 LSA

Latent Semantic Analysis (LSA) is a statistical technique for finding contextual-usage meaning of words in texts. As a practical method for the characterization of word meaning, LSA produces measures of word-word, word-passage and passage-passage relations. It assumes that words that are close in meaning will occur in similar pieces of text.

Initially, the text data is collected for LSA. Then, LSA separates the text data into documents. Each paragraph is generally processed as a separate document like a paragraph being coherent and related. A matrix containing word counts per paragraph is constructed from a large piece of text. A co-occurrence matrix of documents and terms contains the number of word y in the document x. The cell in this matrix corresponding to document x and term y contains the number of times y occurs in x. The number of columns is reduced using Singular Value Decomposition (SVD). In other words, the effect of the common words can be reduced by weighting the values of each cell. Then, the cosine of the angle between the two vectors formed by any two rows is calculated for similarity. They are very similar words if it closes to 1. Else if it closes to 0, they are dissimilar words [42].

2.4.6 Topic analysis with LDA

People are interested in lots of information in many different fields such as articles, images, and survey data. People need to obtain meaningful and organized data. Topic modeling is an important method for analyzing large text data. Latent Dirichlet Allocation (LDA) is one of the popular methods for topic modelling.

Blei et al. first presented LDA model as a graph model for text topic analysis in 2003 [19]. The LDA model was represented as a probabilistic model as shown in Figure 2.6. LDA is a three-level hierarchical Bayesian model [19] which is an unsupervised, statistical method to discover latent semantic topics of large text [43].

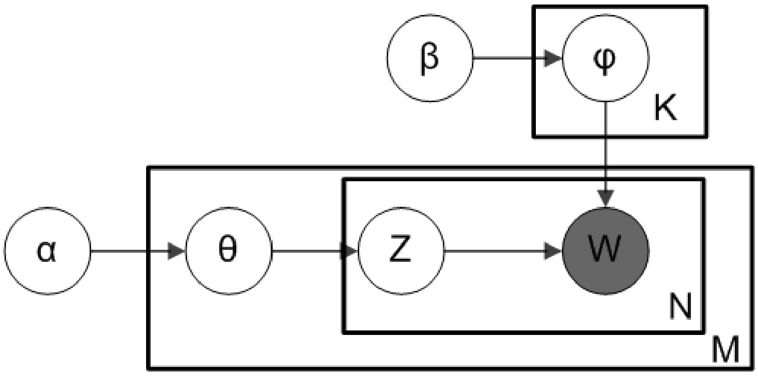


Figure 2.6 Graphical model of LDA.

The basic idea of LDA is that the documents are presented as random mixtures over latent topics where each topic is defined by a distribution over words. LDA supposes that each document consists of mixtures of topics in text modeling. The data of LDA is documents that consist of words. Each document is represented as a mixture of latent topics which are probabilities of different words. The topic probabilities provide an explicit representation of a document. These mixture distributions are assumed to be Dirichlet-distributed random variables which must be inferred from the data. In the generative process; each topic is sample a distribution over words from a Dirichlet prior, each document is sample a distribution over topics from a Dirichlet prior, each word in the document is sample a topic from the document's topic distribution, sample a word from the topic's word distribution and observe the word [19,43,44]. The generative process can be performed in different ways such as inference via Variational Message Passing [19], Expectation Propagation [19], and Gibbs sampling [44].

Representation of documents is generally performed with BoW model [19, 43, 44]. In LDA, the number of topics is *k*. Before performing LDA, the *k* value is determined by user. Number of all words in dictionary is designated as *V*. *M* is number of documents while *N* is number of words in all documents. *A* is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of the Dirichlet prior on the per-topic word distribution, θ_i is the topic distribution for document *i*, ϕ_k is the word distribution for topic *k*, Z_{ij} is the topic for the j^{th} word in document *i*, and W_{ij} is word-level variable.

Gipps Sampling and Expectation Progatation are used to find topics. The following steps of LDA are given.

1. N (Poisson()) is choisen.
2. Q Dirichlet(a) is choisen.
3. Word and topic positions are chosen so that
   3.1. A topic *z* Multinomial(*Q*) is choisen for each of the *N* words.
   3.2. A word w_n for each of the *N* words is choisen from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic $[\beta]_{k \times V}$ $\beta_{ij} = p(w^j = 1 | z^i = 1)$.

Dirichlet distribution is defined over a (*k-1*)-simplex. It takes *k* non-negative arguments which sum to one.

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1} \tag{2.6}$$

The posterior distribution:

$$p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^{N} p(z_n | \theta) p(w_n | z_n, \beta) \tag{2.7}$$

The new topic decomposition of a particular corpus arises from the corresponding posterior distribution of the hidden variables given the *D* observed documents *E w*1*:D*.

$$\left.\begin{aligned} p(\mathbf{w} | \alpha, \beta) &= \int p(\theta | \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n | \theta) p(w_n | z_n, \beta) \right) d^k \theta \\ p(D | \alpha, \beta) &= \prod_{d=1}^{M} \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \theta_d) p(w_{dn} | z_{dn}, \beta) \right) d^k \theta_d \end{aligned}\right\} \tag{2.8}$$

Gibbs Sampler Algorithm

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm. It is a technique for generating random variables from a desired distribution. However it doesn't need to calculate density to do it. Gibbs sampling is particularly well-adapted to sampling the Bayes posterior distribution. It is a technique for generating random variables from a distribution which constitutes a Markov chain, to approximate the joint distribution. However it doesn’t need to calculate density to do it [45].

Gibbs Sampling algorithm steps are as follows.

1. *m* can be iterated *J* times to get $(X_1^j, X_2^j, ..., X_m^j)$, j = 1, 2, … , J.

2. The joint and marginal distributions of generated $(X_1^j, X_2^j, ..., X_m^j)$ converge at an exponential rate to joint and marginal distribution of $(X_1, X_2, ..., X_m)$ as $J \to \infty$.

3. Then the joint and marginal distributions of $(X_1, X_2, ..., X_m)$ can be approximated by the empirical distributions of M simulated values $(X_1, X_2, ..., X_m)$ (j=L+1,…, L+M).

The mean of the marginal distribution may be approximated by $\frac{\sum_{j=1}^{M} X_i^{L+j}}{M}$.

2.4.7 Support Vector Machines (SVMs) and Classification

Boser et al. introduced Support Vector Machine (SVM) in 1992 [46]. Cortes and Vapnik developed SVM for binary classification in 1995 [47]. SVM is used in machine learning, statistics and artificial neural networks [47]. There are very good results were reported to classify text using SVMs [48].

SVM is based on statistical learning theory [48]. SVM is a discriminative classifier which classifies into using optimal hyperplane. One of the main ideas of SVM is that it maximizes margin to find optimal hyperplane. The other is that SVM maps data to high dimensional space for linearly separable problems. Figure 2.7 shows a classification with separating by optimal hyperplane [47]. It gives linearly separable example of SVM.

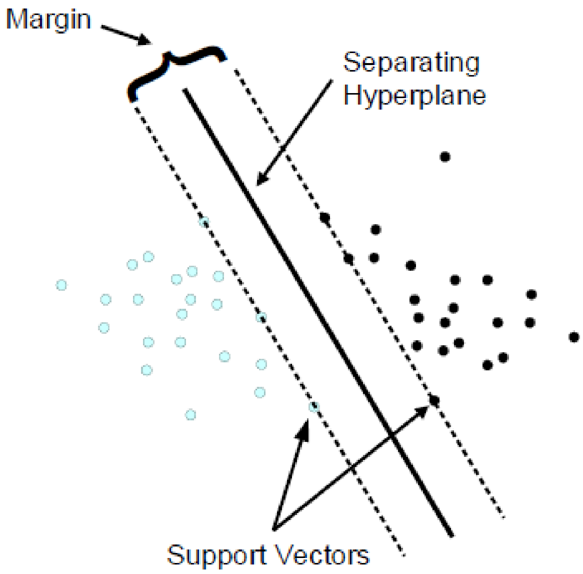


Figure 2.7 Classification (Linear Seperable Case)

2.5 RECOMMENDER SYSTEMS

2.5.1 Overview of Recommender Systems

Recommender system or recommendation system is a subclass of information filtering systems. Recommender systems are widely used on the Web to generate meaningful recommending products and services to users such as recommendation applications in Twitter [49,50, 51].

Finding information searched in the web can be a difficult and time-consuming process. Recommender systems can help to find that information to users [52].

Recommender systems provide to find the user items of their interest. Item is everything to recommend to users. A recommender system focuses on a specific type of item such as what news to read, what music to listen to or what movies to rent [53, 54]. The music, movie and book sites such as Amazon.com, Netflix.com, CDNoW use recommender systems for commercial purposes [55].

Recommender systems gather a list of recommendations in one of three ways through collaborative, content-based or hybrid filtering [52, 55].

2.5.2 Content -based Systems

Content-based recommendation has been performed on information retrieval and machine learning such as text categoration for web pages [56] and books [57]. Some classification algorithms have also been used for content-based recommending, including k-nearest neighbor, decision trees, and neural networks.

Content-based filtering is based on information on the content of items rather than on other users. Due to lack of need to data on other users, this method has an important advantage. The similarity of items is calculated based on the features associated with the compared items [53].

First step is to present items of interest to the user. The user has to provide information on its personal interests on starting to use the system for the profile to be built. The profile includes information about the items of interest, i.e. movies, books, CDs etc. The items generally presents using the tf–idf representation that the term with highest weight occur more often in that document than in other documents more central to the topic of the document. Second step is to compare the user's profile to some reference characteristics to predict whether the user would be interested in an unseen item. The result is a relevance judgment that represents the user's level of interest in that object. If a profile accurately reflects the user preferences, it is of tremendous advantage for the effectiveness of an information access process [52, 58].

2.5.3 Collaborative Systems

Collaborative filtering is a rapidly improving research area [59]. In the early 1990s, collaborative filtering began to arise as a solution for dealing with overload in online information spaces. Automated collaborative filtering systems soon followed, automatically locating relevant opinions and aggregating them to provide recommendations [55]. Collaborative filtering is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behavior of other users in the system. The main assumption in the filtering is based on other users' opinions to provide a reasonable prediction of the active user's preference [55,60].

Several different similarity functions have been proposed and evaluated such as k-nearest neighbor, pearson correlation, constrained pearson correlation, spearman rank correlation, cosine similarity [55].

In collaborative filtering application, user ratings for items which indicate users' interests in an item on a numeric scale are stored in a database. The similarity measures between two user profiles, U and J can be defined by r_{UJ} using pearson correlation or another similarity functions. Once the similarity between profiles has been quantified, it can be used to compute personalized recommendations for users. All users whose similarity is greater than a certain threshold *t* are identified and predictions for an item are computed as the weighted average of the ratings of those similar users for the item, where the weight is the computed similarity. For a given user, other similar users are found whose ratings correlated with the current user. The items rated highly by these similar users are recommended [55].

2.5.4 Hybrid Systems

In order to overcome the difficulties, when collaborative filtering and content-based filtering are combined, several hybrid approaches are performed [61]. The underlying idea is that the content is also taken into account when attempting to identify similar users for collaborative recommendations.

The hybrid recommendation systems are classified into seven categories by Burke in 2002 which are weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level [61].

A weighted hybrid recommender recommends score of item using the result of all of the existing recommendation methods available in the system. The system uses linear formulae in its procedure.

A switching hybrid recommender switches one recommender due to some criterion.

A mixed hybrid recommender merges presentation of multiple ranked lists into one.

A feature combination hybrid recommender uses contributing recommender component and actual recommender component. The features of one source are injected into the source of the other component.

A feature augmentation hybrid recommender is similar to the feature combination hybrids. The difference is that the contributor generates new features.

A cascade hybrid recommender is employed first to produce a coarse ranking of candidates and a second technique refines the recommendation from among the candidate set.

A meta-level hybrid recommender is that contributing and actual recommenders exist but the former one completely replaces the data for the latter one completely [61].

CHAPTER 3

EXPERIMENTAL WORK

3.1 DATA COLLECTION

In the present study, the data used in the design of the recommender system is crawled from Twitter since it is a great knowledge source for research on social media.

Twitter provides an API called SEARCH API for the researches to reach tweets, hashtags, users etc. For this study, the tweets are collected from Twitter users' timeline in batches of 50 tweets per API request. In order to use in this study, train and test data are crawled from Twitter by Twitter SEARCH API.

The crawling procedure for train data is as follows: the first step is to crawl a trend topic and then second step is to crawl a tweet corresponding to the trend topic, after that user of the tweet is obtained, finally all tweets of the user in a day are crawled. This procedure is repeated with Twitter SEARCH API so that more than 60,000 train tweets, corresponding trend topics and users are crawled in order to use for the design of recommender system in this study.

The crawling procedure to get test data is as follows: the first step is to determine high rated topics in Turkey. For this purpose, 5 current topics which are "ak parti", "kpss", "galatasaray", "fem yayınları" and "Barış Manço" are selected. Finding manually the users corresponding to the topics is the second step. The final step is that all tweets of the users are crawled with Twitter SEARCH API. At the end of this procedure, 1,200 test tweets are crawled to measure performance of the recommender system in this study.

3.2 PREPROCESSING

The test tweets are saved into a text file that includes user name, corresponding tweet, whether tweet is a retweet and the date of post consecutively at each line of the file as follows:

“@CimbomHaber_ - Ünal Aysal: Bütün branşlarımızı en az futbol kadar önemsiyoruz. tarih Tue Jul 23 14:25:00 EEST 2013”

“@ChampionCimbom - RT @GalatasaraySK: Galatasaray Sportif A.Ş.’den Duyuru | http://t.co/vK3pMAGr1N tarihThu Jul 18 18:47:25 EEST 2013”

The train tweets are saved into a text file that includes user name, corresponding tweet, and whether tweet is a retweet at each line of the file as follows:

“@ZbydePolat - #Unutmadık seni Barış Manço http://t.co/dvrI51qs”

“@elif_can_12 - RT @13burc: #ondörtşubatgeldiğinde OĞLAK, BALIK, AKREP, ASLAN, BAŞAK :(http://t.co/PgxLSICz”

In the present study, all crawled tweets are preprocessed to transform tweets into BoW model and word document matrix representation. The preprocessing steps of separating the tweets, deleting punctuation, deleting stop words, stemming, building the BoW model, word frequency and word document matrix are explained in the following sections respectively. However, the difference between the preprocessing of test tweets and train tweets is whether the roots of nonexistent words in Zemberek are added to Zemberek. In plain words, the roots of some important words are chosen manually in test tweets since some unimportant words are ignored. On the other hand, all nonexistent words in Zemberek for train tweets are added directly to Zemberek. We had to manually add some words to Zemberek in this way, because, it is difficult to find true roots of a large number of words in the large amount of tweets in our data set.

3.2.1 Separating tweets

This step consists of separating tweets into files, separating user, hashtag, retweet and url of tweets. To separate tweets into files, each tweet in the text file is separated into an individual text file. Then, the user names present in each individual text file are moved to a text file including all user names. The moving procedure is repeated for hashtags, retweets and urls. Hence, we have a file that includes only tweets' hashtags. Similarly, another files are so that one of them indicates whether the tweets are retweet, the other includes only urls in tweets if exists.

3.2.2 Deleting Punctuation

All leading and tailing punctuations are removed in order to simplify processing of tweets.

3.2.3 Deleting Stop Words

In order to process natural language, the stop words should be removed from the text data. There are some studies so that some words can be chosen for any purpose as stop words [62]. In English, there are frequently used words such as the, is, at, which, and on. To improve the language processing, some frequently used words, as “want”, can be considered as stop words [63]. In Turkish, some stop words is listed in Table 3.1.

Table 3.1Turkish stop word samples

a
acaba
altı
ama
ancak
artık
asla
aslında
az

3.2.4 Stemming

The Stemming mechanism is particularly useful in information retrieval [64]. Stemming aims to extract the root of the words [2,64]. In other words, stemming means removing suffixes. For this purpose, there are some algorithms in the literature [64,65].

In this study, stemming is assumed a mechanism for reducing Turkish words in the data. For instance, the stem form of the words “sporcu”and “sporlar” is “spor”. The stem forms of the words in this study are obtained using Zemberek 2 [32] in Java.

Zemberek 2 library doesn’t include all words of crawled tweets. However, Zemberek 2 gives permission to add the words which doesn’t exist in its library in run-time. Hence, firstly any word is controlled whether it exists in the library. If the word isn’t included by the library, it is added into library using codes in run-time. After this process, stemming is performed. If not, stemming is directly performed from the library. For example; “türkiye” isn't found in zemberek. This important word isn't found in Zemberek. The word is added in run-time, then such words “Türkiye'deki” and “Türkiye’yi” can be stemmed and looked for their roots.

3.2.5 BoW Model

BoW model explained in section 2.4.1 is applied for all crawled tweets. A dictionary consists of all words of the tweets. The document represents a row matrix that is frequencies in each tweet of words from the dictionary.

3.2.6 Word Frequency

Word frequency is defined as the number of the word used in all tweets. A text file is considered to save the term frequency of each word. The text file contains all words of the tweets in alphabetical order and number of the words in tweets separated with a colon at each line of the file. The first twelve lines of the text file are given in Table 3.2.

Table 3.2 Word frequency samples

ab:7
abis:2
acil:2
aci:3
acik:1
ad:20
ada:5
adale:2
adalet:3
adam:6
adap:1
aday:17
…

3.2.7 Word Document Matrix

A word document matrix is another text file such that each line of the file contains the number or index of the tweets and the index of the corresponding words in the word frequency file separated by a colon, i.e. 1057:378 indicates 378th word in the word frequency file exists in 1057th tweet.

3.3 EXTRACTING TOPICS

In the present study, more than 63,000 tweets are taken into account. Nearly 62,000 tweets of them are used as training tweets and about 1.200 tweets are used for tests. Among them, each user has more than one tweet. Therefore, all tweets of each user are combined into one tweet file. As a result, the data are 1,780 users and same number of combined tweets. Among them, 1,764 tweets are obtained from tweets of the train user's and the remaining 16 test tweets are obtained from tweets of the test user's. Hence, there are 4 types of data including train tweets, test tweets, train users and test users. The test data is used to evaluate the recommender system. Train data is used to find similar tweets and users.

4 types of data are operated using the generative modeling approach LDA, and cosine similarity of k-means. LDA is used to extract topics and k-means is used to cluster tweets and users. The determining of the number of topics is difficult while performing LDA. The tests are experimented to find the suitable number of topics. If we prefer 10 topics, different topics mix with each other. For example, “KPPS topic” and “sports topic” are found in forth topic. So the number of topics is not high enough for this clustering. Hence 40 topics for 62,000 train tweets, 17 topics for 1,200 test tweets, 7 topics for 16 test users and 40 topics for 1,700 train users are preferred. The other challenge is to determine the number of clusters of tweets. Hence 40 clusters for 62,000 train tweets, 17 clusters for 1,200 test tweets, 7 clusters for 16 test users and 40 clusters for 1,700 train users are preferred.

In Figures 3.1 to 3.6, the number of tweets per cluster is shown for test tweets that are experimented using cosine similarity and k-means with 5, 10, 15, 17, 20 and 30 clusters, respectively. Figures 3.7 to 3.14 show the number of tweets per cluster for training tweets that are experimented using cosine similarity and k-means with 5, 10, 15, 20, 30, 40, 50 and 100 clusters, respectively. Figures 3.15 to 3.17 show the number of tweets per cluster for test users that are experimented using cosine similarity and k-means with 5, 7 and 10 clusters, respectively. Figures 3.18 to 3.22 show numbers of tweets per cluster for training users that are experimented using cosine similarity and k-means with 5, 10, 20, 30 and 50 clusters respectively.

In summary, Figures 3.1 to 3.22 show that different number of topics and different number of clusters lead to different results. Hence, finding suitable number of topics and clusters are very important for successful clustering. If we increase the number of topics, the distribution among topics is more balanced. If we increase the number of topics even more, in this case, the distribution among topics damages. Similarly, if we increase the number of clusters, distribution among clusters is more balanced. If we increase the number of cluster too much, in this case the distribution among clusters damages. An ideal number of topics and an ideal number of clusters are found with necessary number of tests.

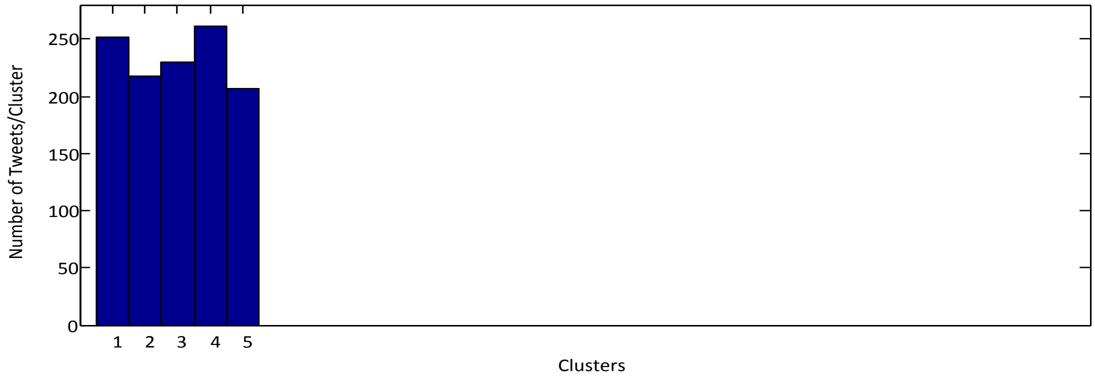


Figure 3.1 Clustering with cos similarity of k-means with 5 topics of test tweets

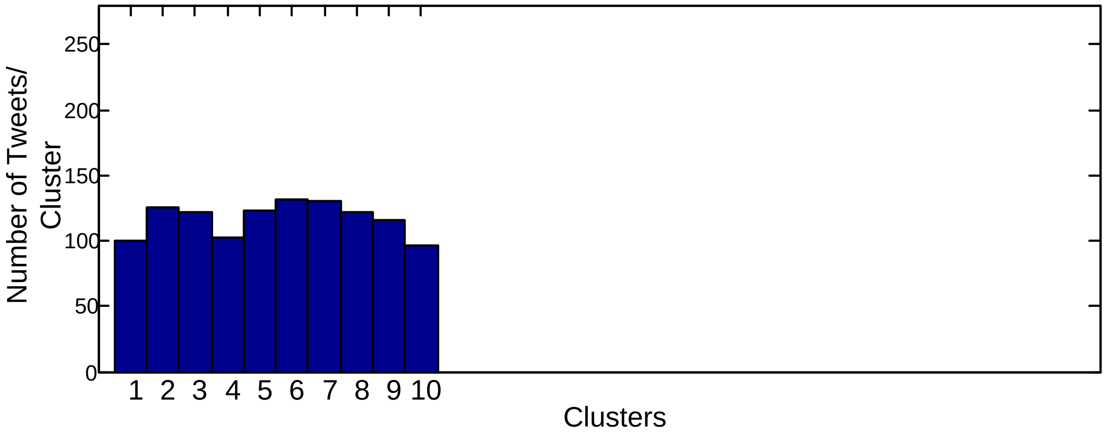


Figure 3.2 Clustering with cos similarity of k-means with 10 topics of test tweets

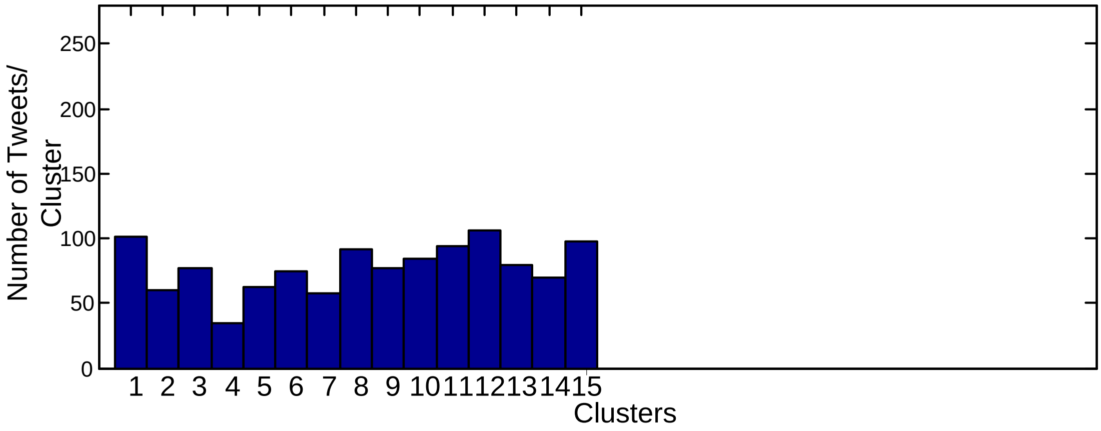


Figure 3.3 Clustering with cos similarity of k-means with 15 topics of test tweets

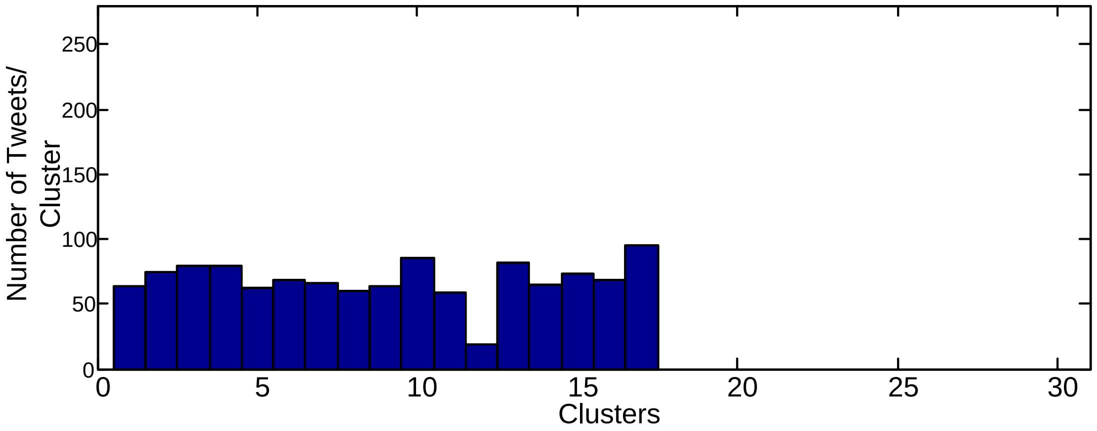


Figure 3.4 Clustering with cos similarity of k-means with 17 topics of test tweets

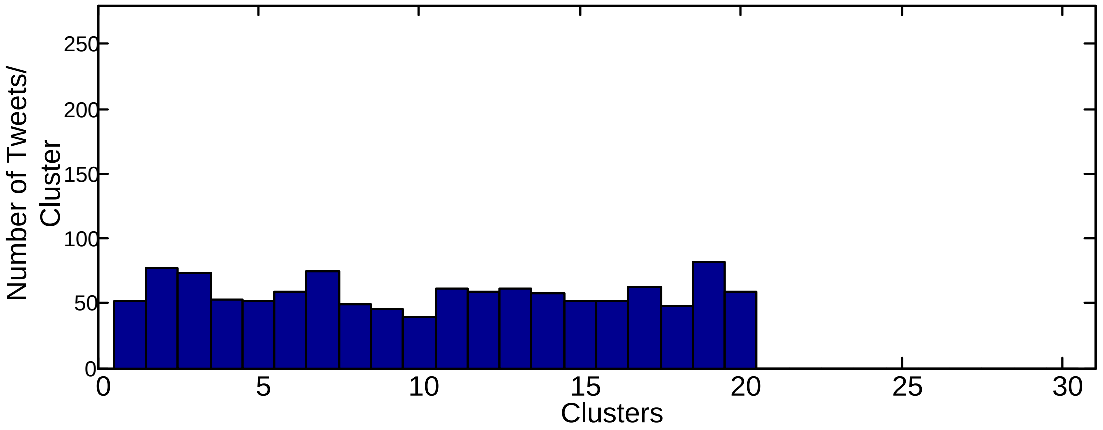


Figure 3.5 Clustering with cos similarity of k-means with 20 topics of test tweets

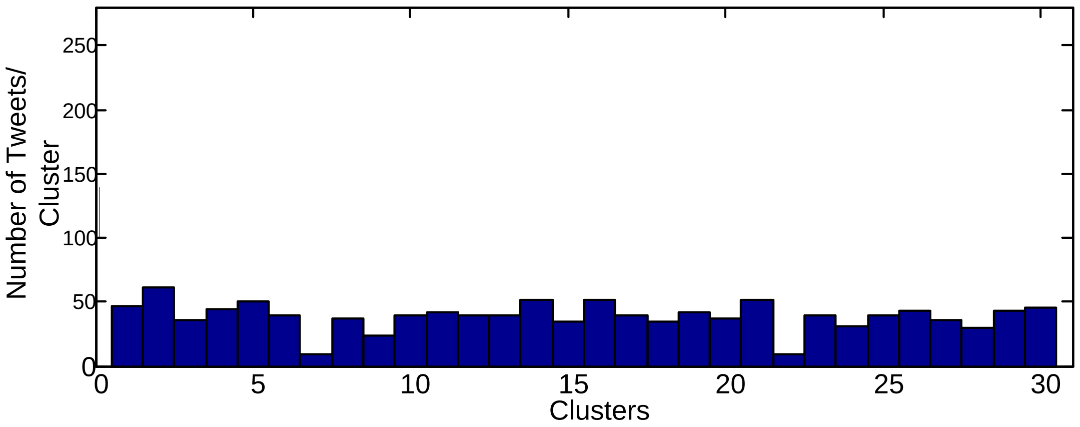


Figure 3.6 Clustering with cos similarity of k-means with 30 topics of test tweets

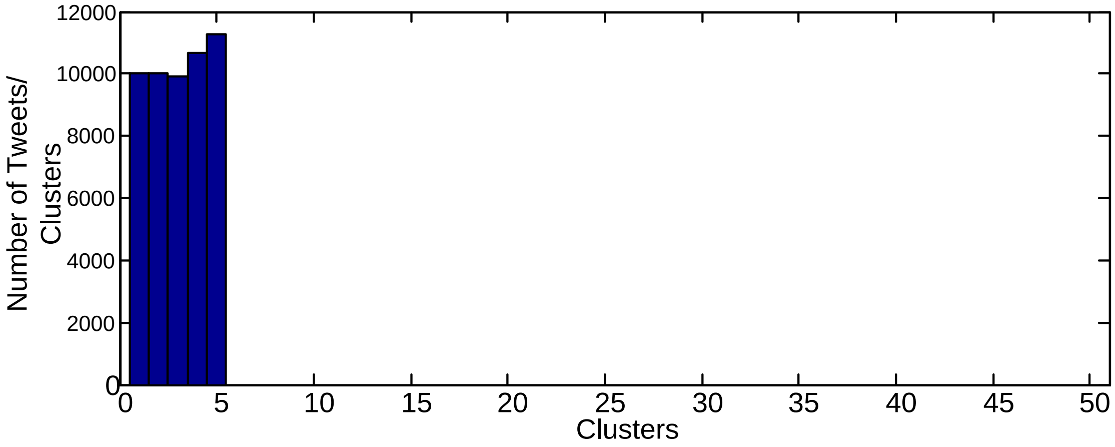


Figure 3.7 Clustering with cos similarity of k-means with 5 topics of train tweets

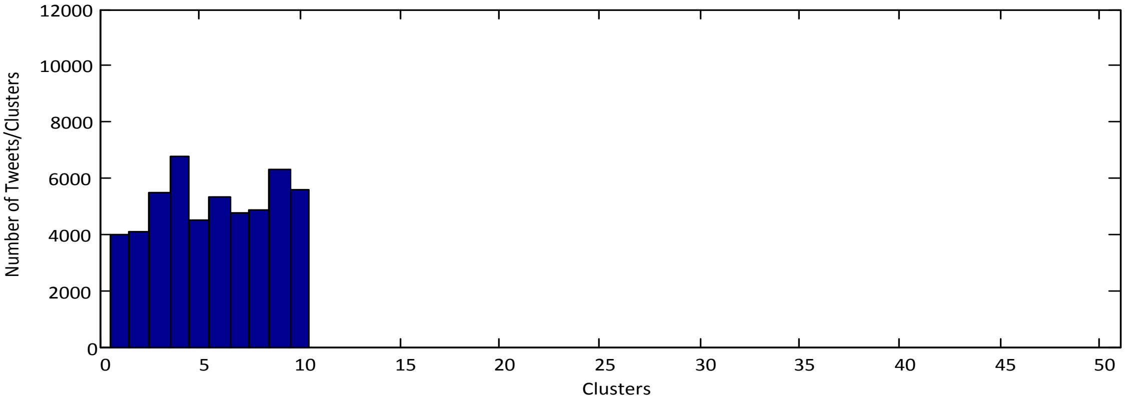


Figure 3.8 Clustering with cos similarity of k-means with 10 topics of train tweets

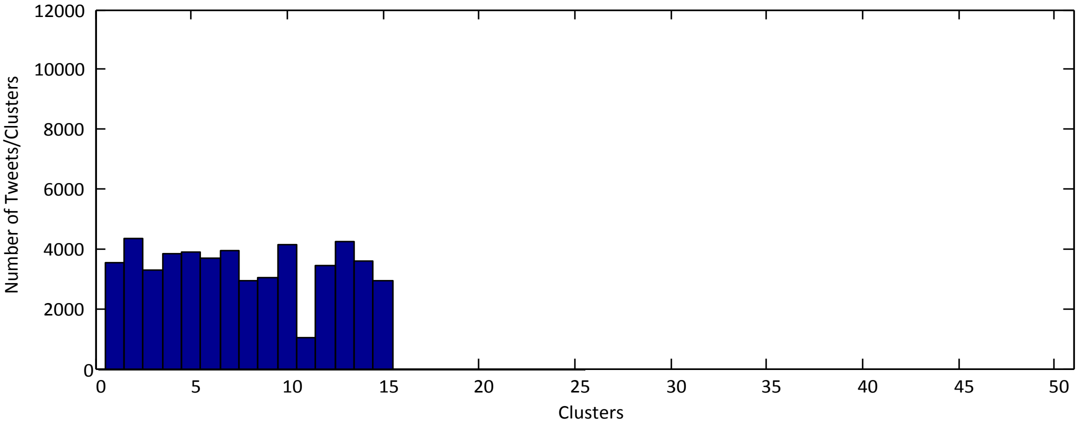


Figure 3.9 Clustering with cos similarity of k-means with 15 topics of train tweets

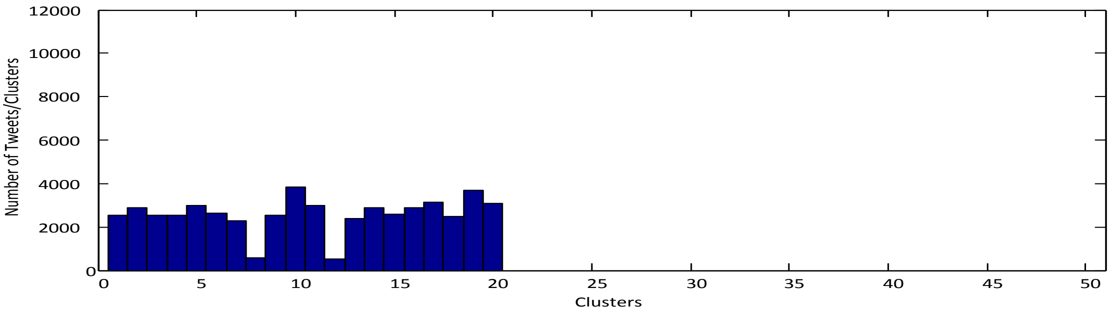


Figure 3.10 Clustering with cos similarity of k-means with 20 topics of train tweets

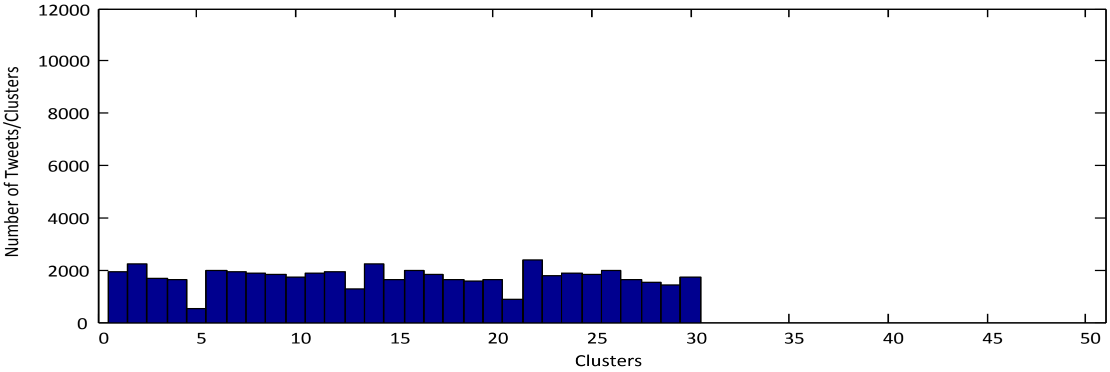


Figure 3.11 Clustering with cos similarity of k-means with 30 topics of train tweets

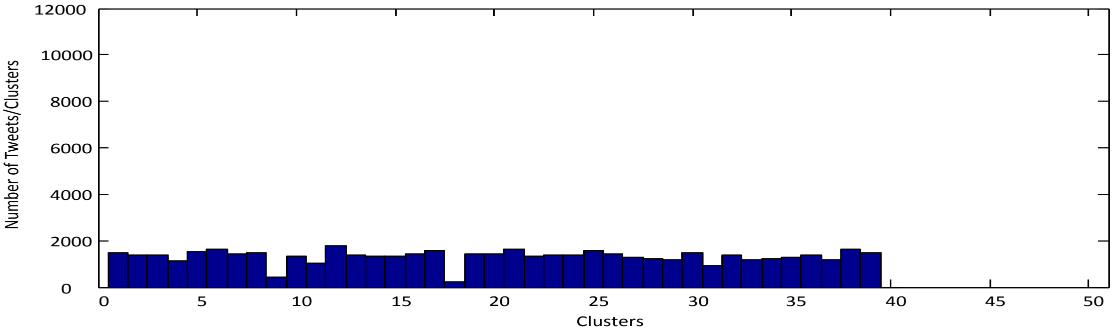


Figure 3.12 Clustering with cos similarity of k-means with 40 topics of train tweets

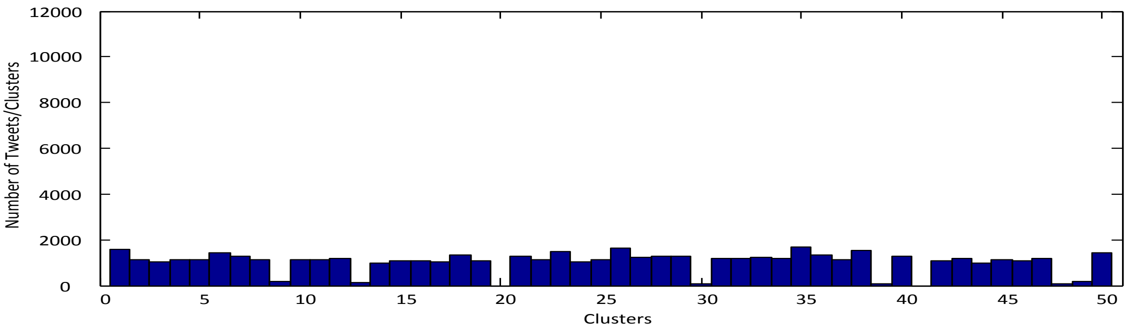


Figure 3.13 Clustering with cos similarity of k-means with 50 topics of train tweets

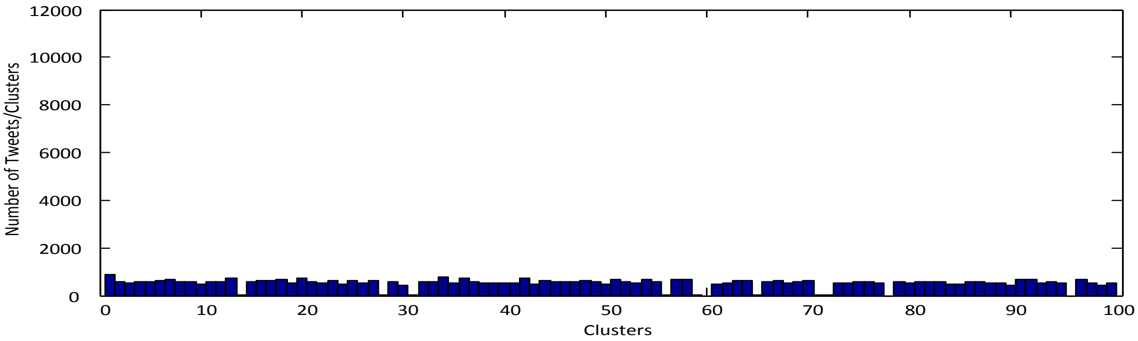


Figure 3.14 Clustering with cos similarity of k-means with 100 topics of train tweets

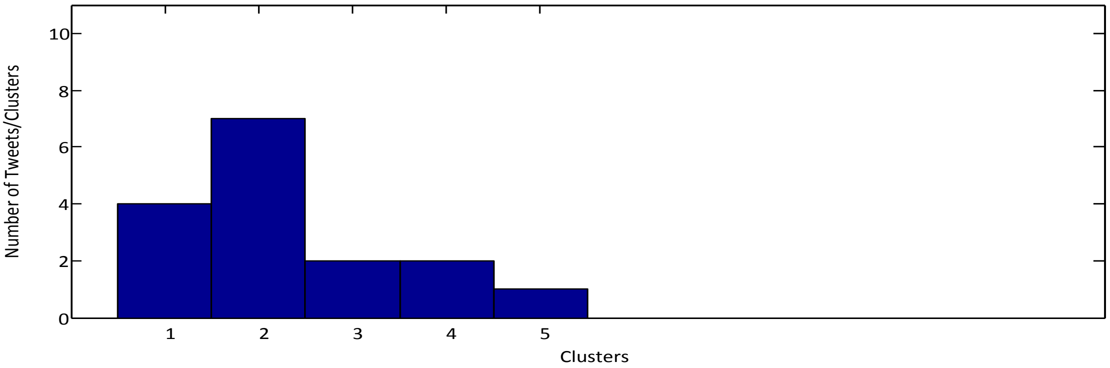


Figure 3.15 Clustering with cos similarity of k-means with 5 topics of test users

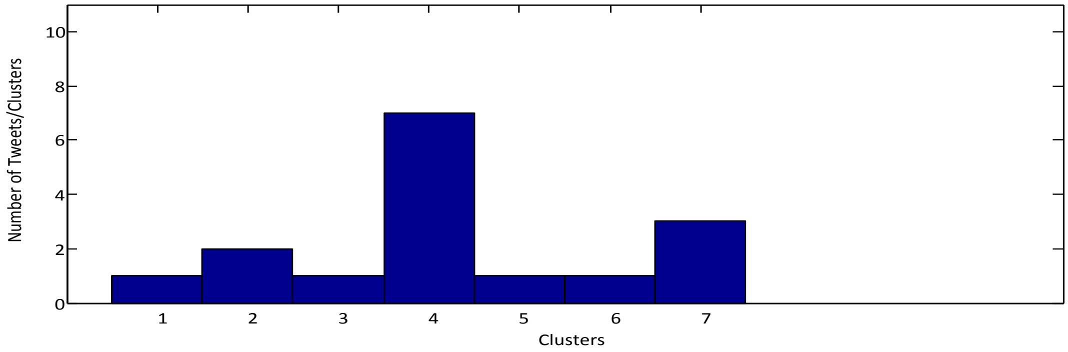


Figure 3.16 Clustering with cos similarity of k-means with 7 topics of test users

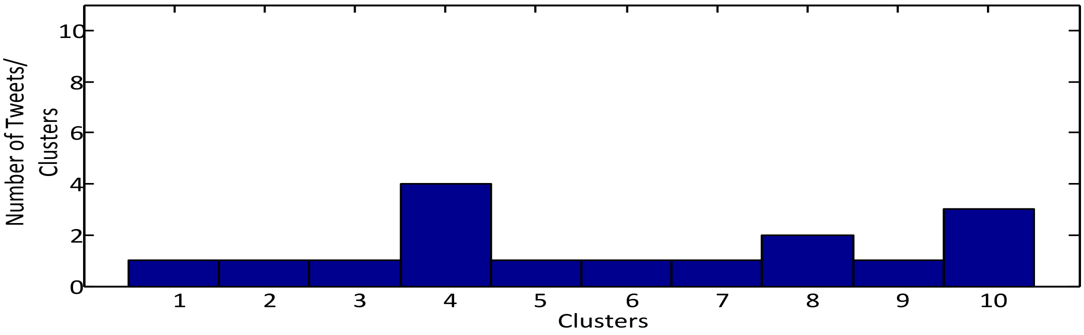


Figure 3.17 Clustering with cos similarity of k-means with 10 topics of test users

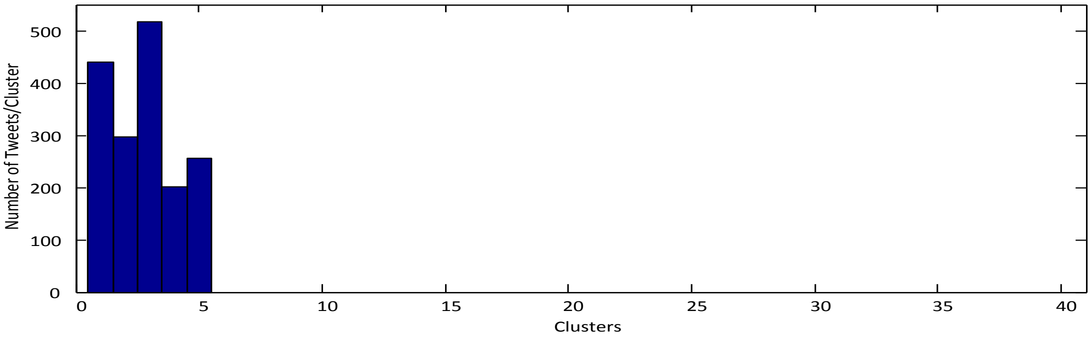


Figure 3.18 Clustering with cos similarity of k-means with 5 topics of train users

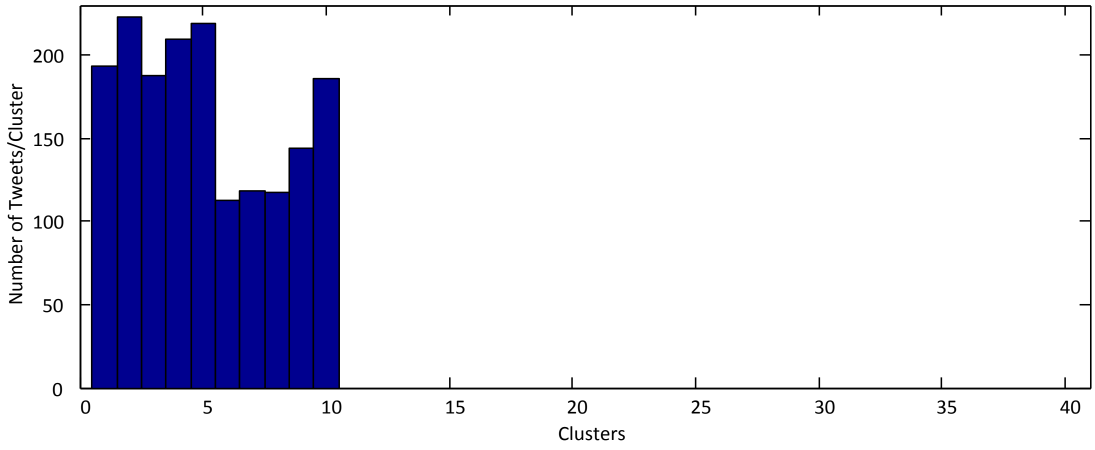


Figure 3.19 Clustering with cosine similarity of k-means with 10 topics of train users

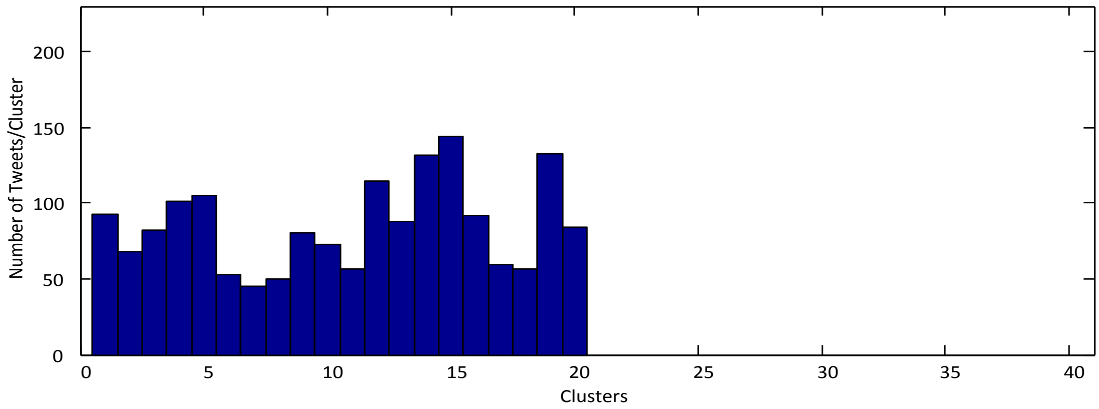


Figure 3.20 Clustering with cos similarity of k-means with 20 topics of train users

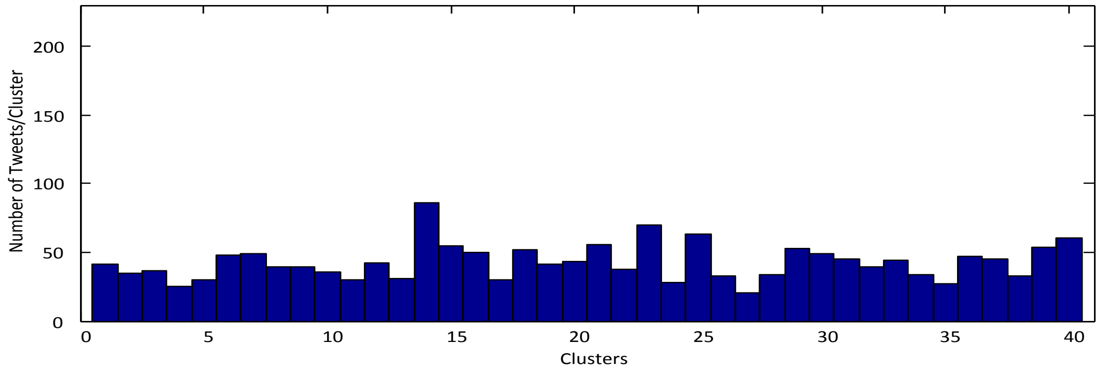


Figure 3.21 Clustering with cos similarity of k-means with 40 topics of train users

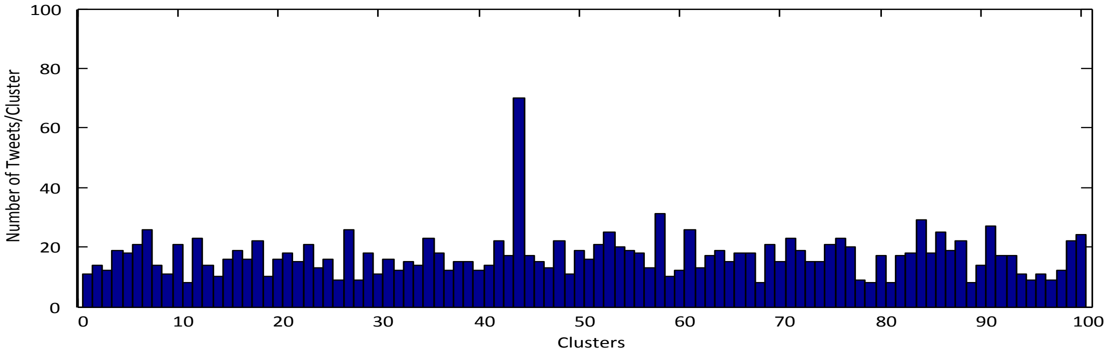


Figure 3.22 Clustering with cos similarity of k-means with 100 topics of train users

First twenty words in five topics of train tweets are given in Table 3.3. It is clear that topic 4 is related with Galatasaray football team, topic 6 is related with AÖF and yds exam, topic 13 is related with kpss exam, topic 25 is related with Akparti political party and topic 28 is related with politics. It is obvious that the clustering algorithm successfully extracts the topics such as Akparti, kpss and Galatasaray etc which are initially determined at the beginning of clustering process.

Table 3.3 Instances of train tweets topics

Topic 4	**Topic 6**	**Topic 13**	**Topic 25**	**Topic 28**
Bekle	sınav	Kpss	Yol	Gel
Kal	Allah	Soru	Yıl	El
Kırmızı	Hayır	Çöz	Bugün	Bil
Burak	Tek	Tıkla	Cevap	Sene
Gol	Dil	Torpil	Aç	Inan
tekBüyükGalatasaray	Hayat	Video	Bruma	Mısır
Sarı	Yayın	Akademi	Hal	Başkan
Tweet	Başarı	Genel	Site	Anayasa
Detay	Büyük	Iyi	Atatürk	Chp
Tertemiztarihiyle	Aöf	Işbirlik	Mücadele	Köprü
Tribün	Ders	Yayımla	Dış	Lys
Kes	Belge	Bahane	Kriz	To
Temmuz	Mühendis	Etiket	Akparti	Deplasman
Lütfen	Yalnız	Hesap	Ayrıl	Facebook
Sahur	Aile	Kör	İnşaat	Darbe
Salih	Intihar	Psikoloji	Kalbi	Haydi
Üçbüyükyok	Tu	Ait	Kontrat	Kaybet
Şekil	Vefat	Coğrafya	Onur	Kayıt
Anla	Yds	Gstv	Savaş	Kol
davran	baş	kariyer	Sebep	millet

3.4 HASHTAG AND URL SIMILARITY

Hashtag is a word or phrase that is a topic determined by users as mentioned in Section 2.2. Since the topic of tweets can be understood from the corresponding hashtag, it is very important to find similar tweets and users.

If the tweets have not different aims like spams, it can be considered that url gives an idea about topics of tweets. Hence, the url in tweets helps to distinguish tweets from each other.

In this study, it is assumed that tweets that contain one or more number of common hashtags are similar to each other and tweets that contains one or more number of common urls are similar to each other.

While cosine similarity of tweets is calculated with k-means, hashtag similarity and url similarity of tweets are calculated with the procedure as follows. Hashtag similarity is assumed whether the tweets include at least one same hashtag with each other. Similarly, url similarity is assumed whether tweets include at least one same url with each other.

For example, the following two tweets are similar because they contain the same hashtag as “#AkTakipBaşlıyor”.

@MuratCubuk_ - RT @Semihbekta_s: Teroru bitirmek icin partisinin gelecegini dusunmeden yola cikan basbakanimiz icin.. #AkTakipBaşlıyor tarih Sun Jul 21 15:08:05 EEST 2013

@MuratCubuk_ - Yer sofrasında oturmayı bilmeyen lider iktidar olamaz bu ülkede #AkTakipBaşlıyor”@zfrcbkc @MacitMahmut tarih Sun Jul 21 15:05:44 EEST 2013

The following two tweets are similar because of containing same url as “http://t.co/WQe4MXRAC7”.

@kpsscafe - http://t.co/WQe4MXRAC7 İlk Arke Akademi işbirliği ile 2013 KPSS Sorularının Videolu Çözümleri http://t.co/sMbNbjMhFC tarih Sun Jul 21 13:43:33 EEST 2013

@kpsscafe - http://t.co/WQe4MXRAC7 Uzman Kariyer Akademi işbirliği ile 2013 KPSS Soruları ve Çözümleri http://t.co/8GYQyPeqNA tarih Sun Jul 21 13:12:03 EEST 2013

3.5 RECOMMENDER DESIGN

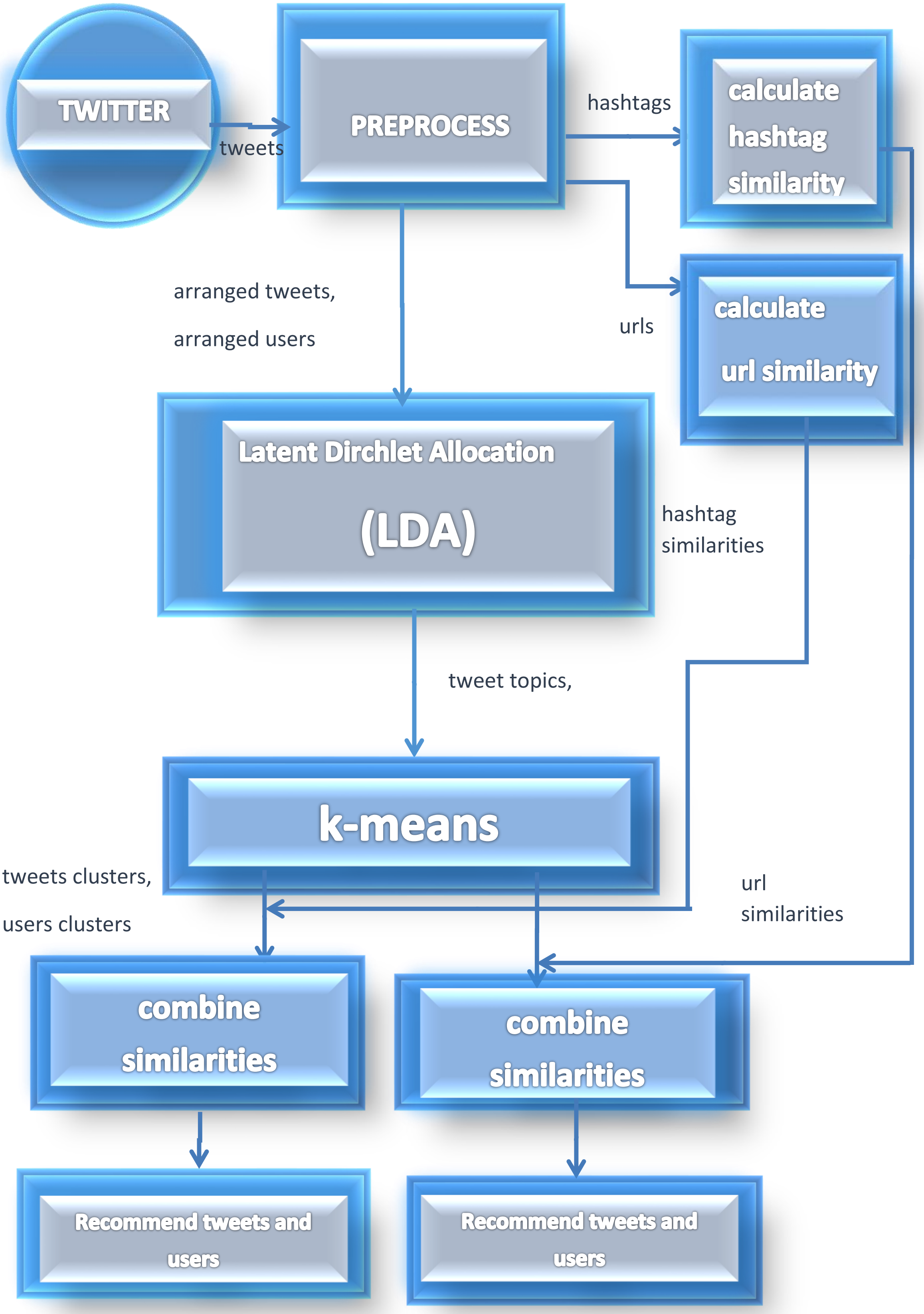


Figure 3.23 Structure of Recommender

In this study, a recommender system that recommends tweets and users is designed for Turkish language. For this purpose, tweet-tweet similarity and user-user similarity are calculated.

Tweet-tweet similarity means that is calculated hashtag similarity of similar tweets that are found with tweet text similarity and calculated url similarity of similar tweets that are found with tweet text similarity. In other words, having common hashtags from the tweets in the same cluster are similar tweets and having common urls from the tweets in the same cluster are similar tweets.

User-user similarity means that is calculated hashtag similarity of similar users that are found with user text similarity and calculated url similarity of similar users that are found with user text similarity. In other words, having common hashtags from the users in the same cluster are similar users and having common urls from the urls in the same cluster are similar urls.

They are formalized as followed.

Tweet-Tweet Similarity = Tweet Text Similarity + Hashtag Similarity
Tweet-Tweet Similarity = Tweet Text Similarity +URL Similarity
User-User Similarity = User Text Similarity + Hashtag Similarity
User-User Similarity = User Text Similarity + URL Similarity

Hence, the system recommends 63,000 tweets and 1,780 users to each other using tweet-tweet similarity and user-user similarity. The results of the recommender system are promising.

CHAPTER 4

ANALYSIS OF THE RESULTS AND CONCLUSION

In this thesis work we developed a recommender system for Turkish Twitter users and Turkish tweets. Our system uses LDA to extract the topics of tweets. A number of preprocessing steps such as stop word removal and stemming using the Zemberek library is used. After extracting topics from tweets, tweets are clustered using their topic distributions. Tweets that fall in the same cluster are determined as similar and recommended to each other. Different number of topics and clusters are tried in the experiments. The number of topics extracted and the number of clusters used affects the results, hence these should be determined based on the application and the size of the data. We apply the same process to the whole set of tweets from users, so that we can recommend users to other users, instead of individual tweets. Our results seem promising.

In Table 4.1, distribution of the tweets for each cluster to 4 topics is given. Test data is divided into 18 clusters and Cluster 0 consists of unclustered tweets. The data consists of siyaset, kpss, takım and fem topics. In Table 4.2, the distribution of tweets into clusters is shown in percent and corresponding distribution graph is given in Figure 4.1. Percent distribution of tweets in each cluster is tabulated in Table 4.3 and corresponding bar graph is shown in Figure 4.2.

The topics KPSS and FEM are successfully distributed where nearly half of KPSS topic, about 55%, is found in Cluster 1 and Cluster 5 and 62.96% of FEM topic is found in three clusters. The remaining topics, named Siyaset and Takım are regularly distributed into clusters. On the other hand, 66 tweets from 1233 tweets are unclustered. Therefore, 5.35% of them are not clustered. Due to the amount of unclustered tweets is 5.35% of all tweets, the system is successfully clustered.

Table 4.1 The distribution of the tweets for each cluster to 4 subjects

	Siyaset	KPSS	Takım	Fem	Total Number of Tweets
Cluster 0	24	9	33	0	66
Cluster 1	3	40	15	0	58
Cluster 2	20	13	35	0	68
Cluster 3	15	3	42	0	60
Cluster 4	10	0	51	1	62
Cluster 5	12	46	11	3	72
Cluster 6	11	2	42	1	56
Cluster 7	6	0	72	2	80
Cluster 8	5	0	66	1	72
Cluster 9	21	7	31	1	60
Cluster 10	5	0	67	10	82
Cluster 11	8	2	66	0	76
Cluster 12	16	0	60	0	76
Cluster 13	3	12	60	2	77
Cluster 14	13	2	46	4	65
Cluster 15	30	0	32	0	62
Cluster 16	25	17	33	2	77
Cluster 17	31	3	30	0	64
Total	**258**	**156**	**792**	**27**	**1233**

Table 4.2 The percent distribution of the tweets for each cluster to 4 topics

	Siyaset	KPSS	Takım	Fem
Cluster 0	9.30%	5.77%	4.17%	0.00%
Cluster 1	1.16%	25.64%	1.89%	0.00%
Cluster 2	7.75%	8.33%	4.42%	0.00%
Cluster 3	5.81%	1.92%	5.30%	0.00%
Cluster 4	3.88%	0.00%	6.44%	3.70%
Cluster 5	4.65%	29.49%	1.39%	11.11%
Cluster 6	4.26%	1.28%	5.30%	3.70%
Cluster 7	2.33%	0.00%	9.09%	7.41%
Cluster 8	1.94%	0.00%	8.33%	3.70%
Cluster 9	8.14%	4.49%	3.91%	3.70%
Cluster 10	1.94%	0.00%	8.46%	37.04%
Cluster 11	3.10%	1.28%	8.33%	0.00%
Cluster 12	6.20%	0.00%	7.58%	0.00%
Cluster 13	1.16%	7.69%	7.58%	7.41%
Cluster 14	5.04%	1.28%	5.81%	14.81%
Cluster 15	11.63%	0.00%	4.04%	0.00%
Cluster 16	9.69%	10.90%	4.17%	7.41%
Cluster 17	12.02%	1.92%	3.79%	0.00%
Total Percent of Each Topic	100%	100%	100%	100%

It is obvious that the maximum ratio of Cluster 1, 5 and 10 in Figure 4.1 are KPSS and FEM topics which means that the corresponding clusters provides correct tweet and user recommendation.

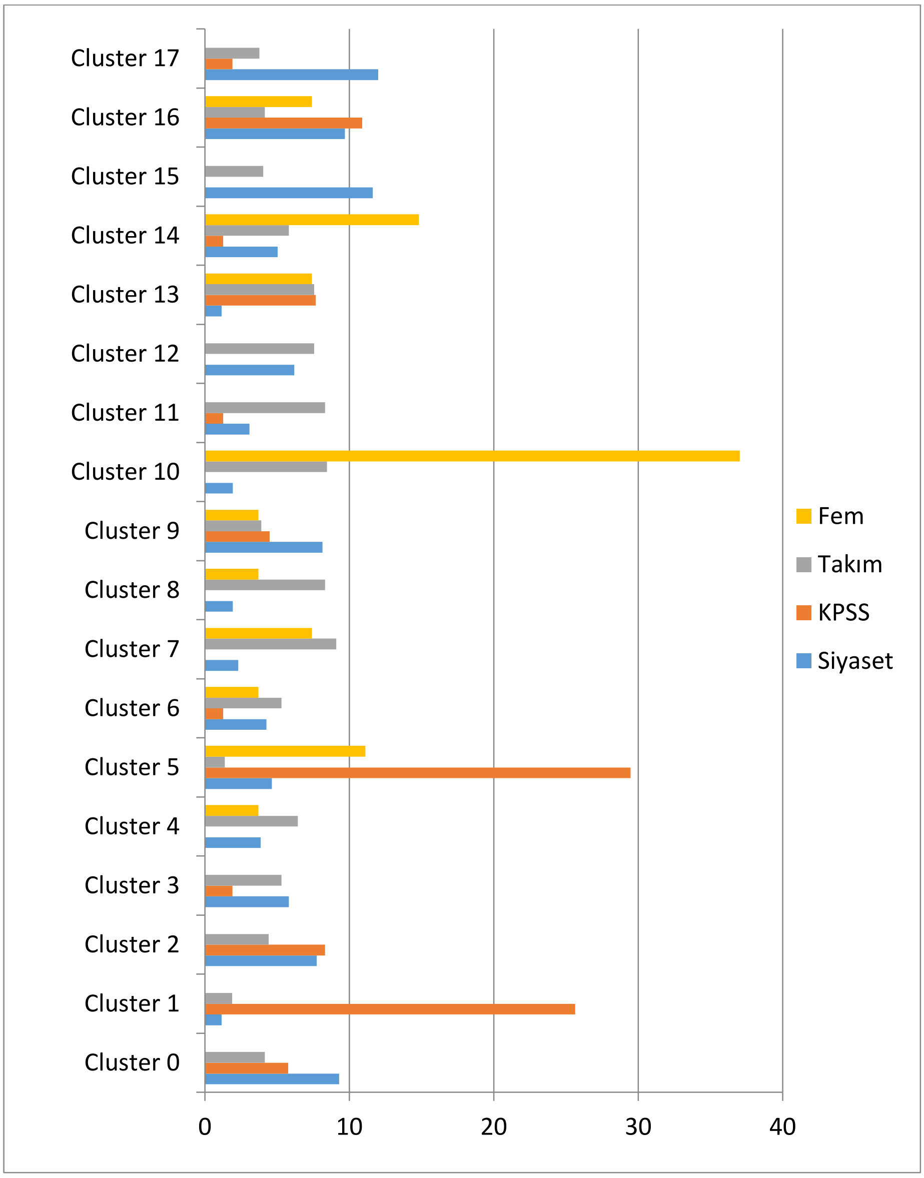


Figure 4.1 Bar graph of the percent distribution of the tweets for each cluster to 4 topics

Table 4.3 Distribution of tweets in each cluster in percent

	Siyaset	KPSS	Takım	Fem	Total
Cluster 0	48.35%	29.99%	21.66%	0.00%	100%
Cluster 1	4.05%	89.35%	6.60%	0.00%	100%
Cluster 2	37.81%	40.64%	21.55%	0.00%	100%
Cluster 3	44.59%	14.75%	40.67%	0.00%	100%
Cluster 4	27.65%	0.00%	45.93%	26.42%	100%
Cluster 5	9.97%	63.23%	2.98%	23.82%	100%
Cluster 6	29.30%	8.81%	36.44%	25.45%	100%
Cluster 7	12.35%	0.00%	48.29%	39.35%	100%
Cluster 8	13.87%	0.00%	59.63%	26.50%	100%
Cluster 9	40.21%	22.16%	19.33%	18.29%	100%
Cluster 10	4.09%	0.00%	17.83%	78.08%	100%
Cluster 11	24.38%	10.08%	65.53%	0.00%	100%
Cluster 12	45.01%	0.00%	54.99%	0.00%	100%
Cluster 13	4.88%	32.27%	31.78%	31.07%	100%
Cluster 14	18.70%	4.76%	21.56%	54.98%	100%
Cluster 15	74.21%	0.00%	25.79%	0.00%	100%
Cluster 16	30.13%	33.88%	12.96%	23.03%	100%
Cluster 17	67.78%	10.85%	21.37%	0.00%	100%

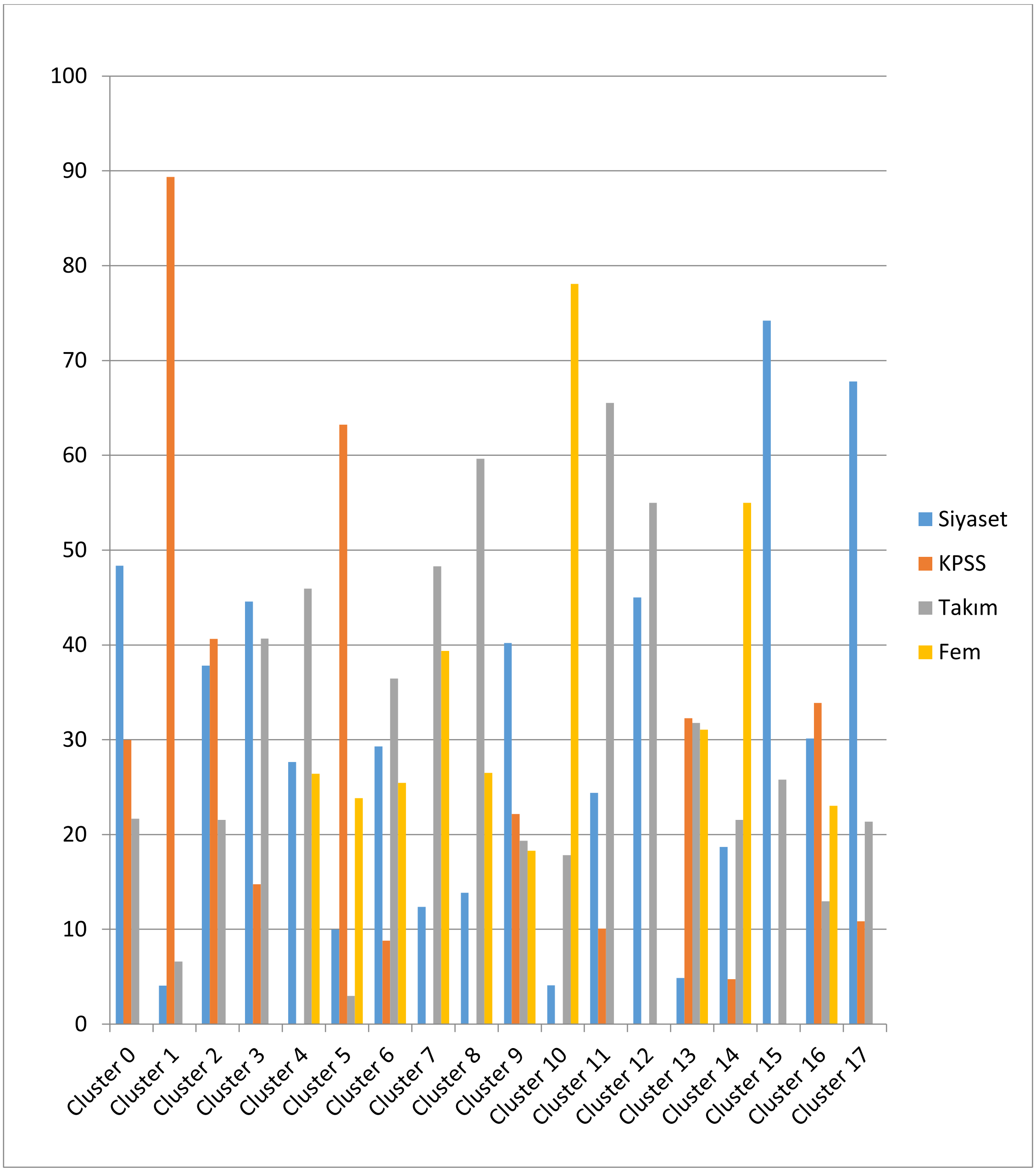


Figure 4.2 Bar graph of distribution of tweets in each cluster in percent

4.1 CHALLENGES

Turkish Twitter users use a different language that may be very different from the accepted correct Turkish language. Analyzing and understanding language of Turkish Twitter users is very hard. It may be a separate article topic for academic research. The language used in tweets with its own rules, wide spread spelling errors or made-up words make text analysis very difficult. Tweets often have words written with false spelling. For example, a post word or a pre word has been written with the same characters together such as ‘gollllllll !!’ and ‘hadiiiiii’. These words are meaningless for our algorithm. We use Zemberek library to find root of words. But root of some words are unobtainable in Zemberek library. So, we create a text file that consists of root of important words manually. In addition, some important words such as “Türkiye” aren’t found in Zemberek. Such case shows that Zemberek library doesn't contain some important words. Here, Zemberek seems to have some deficiency. Zemberek software only gives permission for adding words temporarily when Zemberek is running. When it exits, the added words are deleted. Therefore, the roots of important words in the text file are added to Zemberek in run-time.

The other problem is the blank tweets obtained from preprocessing. In this case, if the tweets have common hashtags or urls, they are accepted as similar.

4.2 FUTURE WORK

Language of Turkish Twitter users is still a problem to be solved. Recognizing a word like ‘gollllllll !!’ as ‘gol!’ will be an important contribution to work with Turkish tweets.

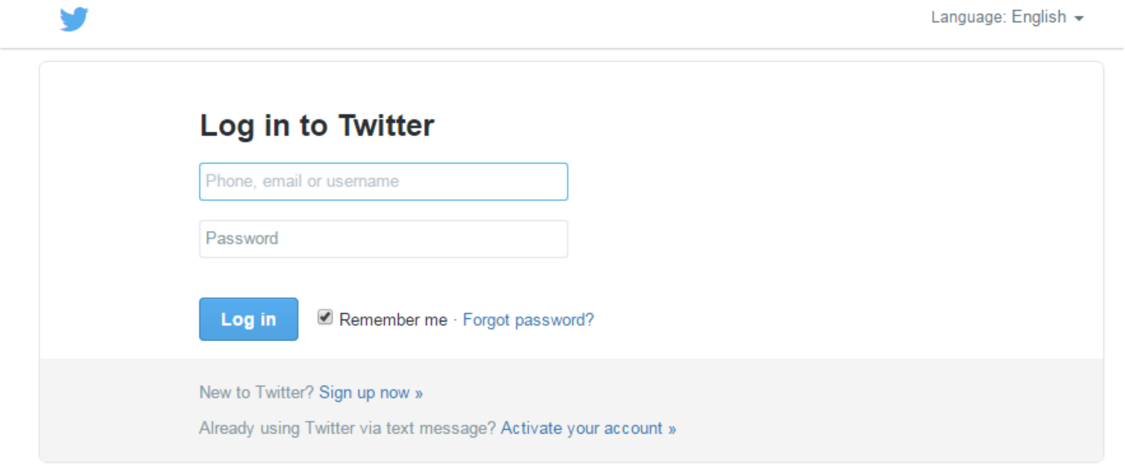
Since the launch of Twitter in 2007, micro-blogging became highly popular and researchers focused to investigate Twitter’s information propagation patterns or analyzed structures of the Twitter network to identify influential users.

As a result, our recommendation system has been promising. There are many possibilities of the system to be conducted in the future. In the future, we will focus on a recommender system using content-based filtering and collaborative-based filtering together for Turkish tweets.

APPENDIX

A twitter application can be created by following the steps given as follows:

1. Login in Twitter application page, https://twitter.com/login?lang=en, as shown in Figure A.1.



Language: English

Log in to Twitter

Phone, email or username

Password

Log in

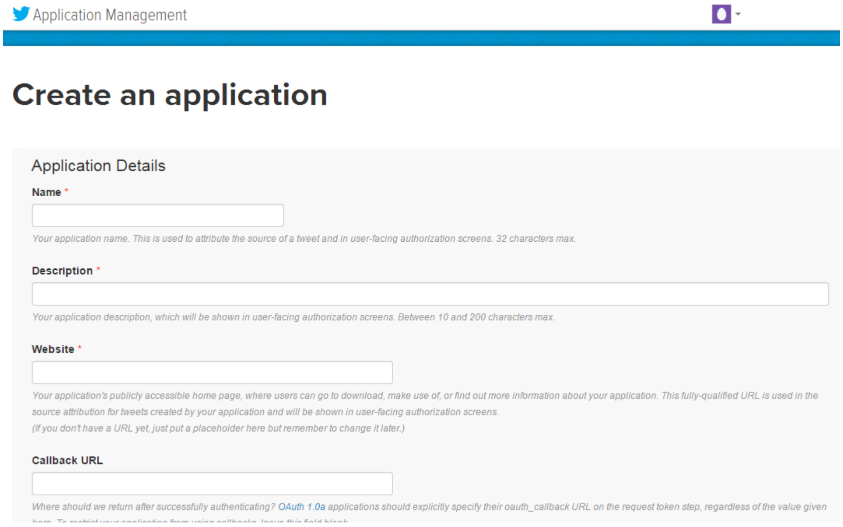
☑ Remember me · Forgot password?

New to Twitter? Sign up now »

Already using Twitter via text message? Activate your account »

Figure A.1 Twitter Login Page.

2. Click “Create New App” button in page https://apps.twitter.com/
3. Fill in the form shown in Figure A.2.



Application Management

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

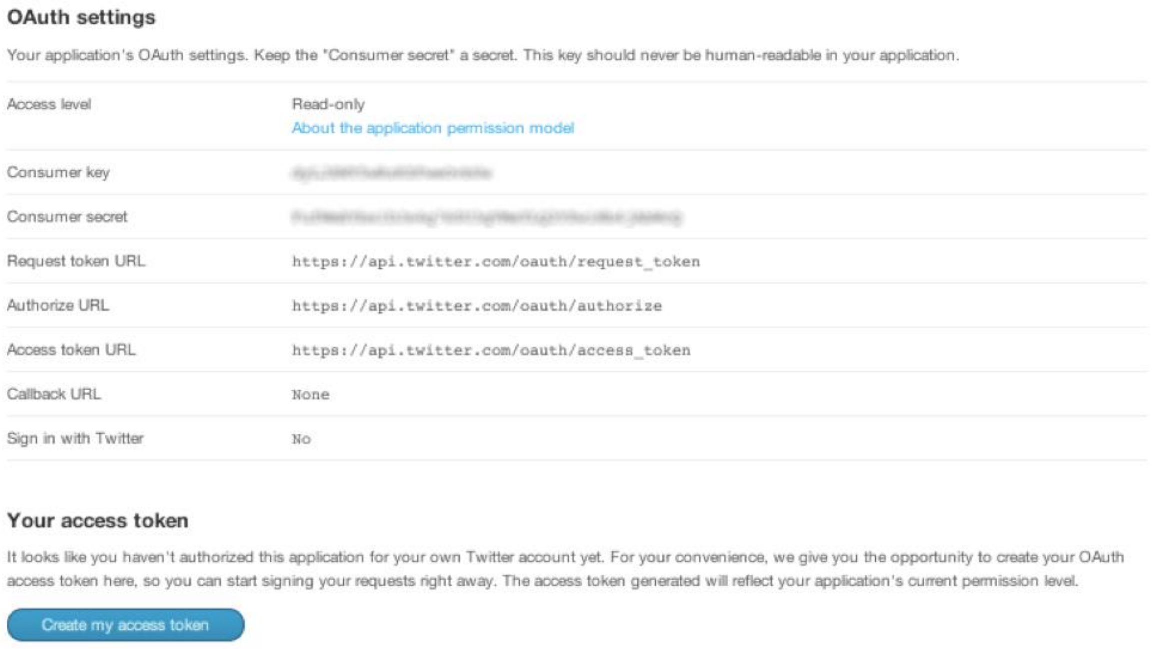
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Figure A.2. A screenshot for creating an application.

4. Create a new application
5. Enter application name.
6. Define the application.
7. Click “Create your Twitter application ” button, after filling the details.
8. The settings come into screen as shown in Figure A.3.



OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

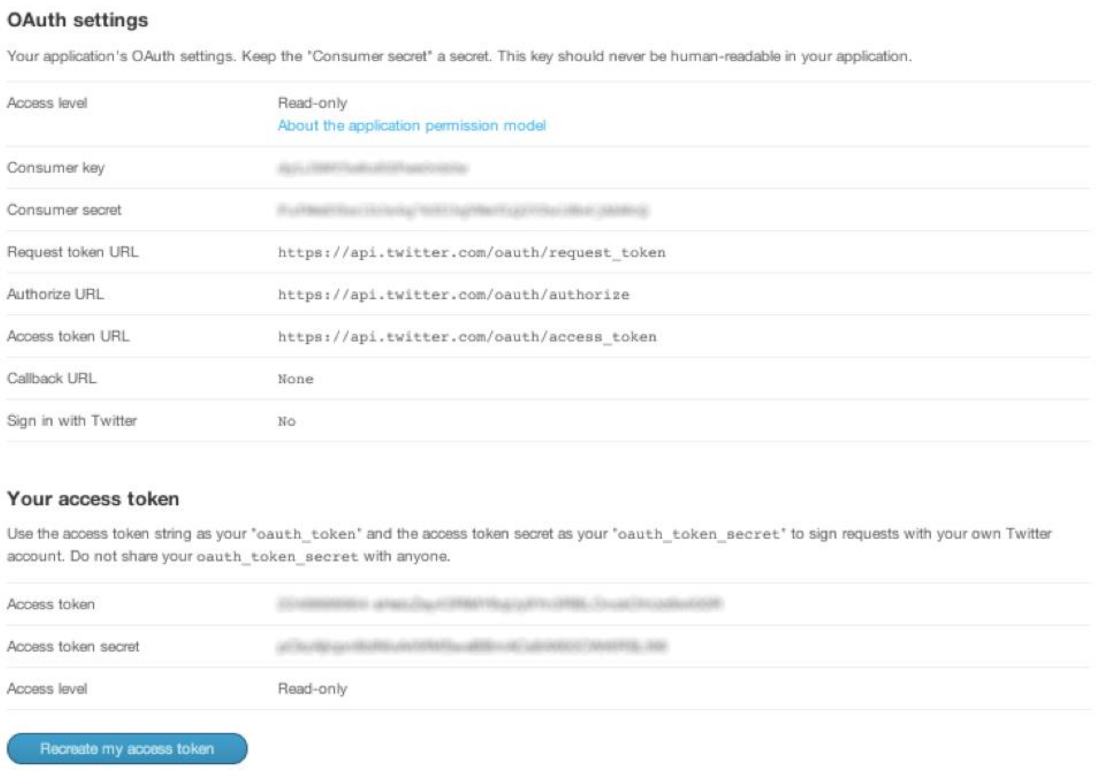
Access level	Read-only About the application permission model
Consumer key	[redacted]
Consumer secret	[redacted]
Request token URL	`https://api.twitter.com/oauth/request_token`
Authorize URL	`https://api.twitter.com/oauth/authorize`
Access token URL	`https://api.twitter.com/oauth/access_token`
Callback URL	None
Sign in with Twitter	No

Your access token

It looks like you haven't authorized this application for your own Twitter account yet. For your convenience, we give you the opportunity to create your OAuth access token here, so you can start signing your requests right away. The access token generated will reflect your application's current permission level.

Figure A.3 The OAuth settings.

9. Click “Create the Access token” button.
10. The settings come into screen as shown in Figure A.4.
11. Finally, oauth_token and oauth_token_secret are generated to connect Twitter .



OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

Access level	Read-only About the application permission model
Consumer key	[redacted]
Consumer secret	[redacted]
Request token URL	`https://api.twitter.com/oauth/request_token`
Authorize URL	`https://api.twitter.com/oauth/authorize`
Access token URL	`https://api.twitter.com/oauth/access_token`
Callback URL	`None`
Sign in with Twitter	`No`

Your access token

Use the access token string as your "`oauth_token`" and the access token secret as your "`oauth_token_secret`" to sign requests with your own Twitter account. Do not share your `oauth_token_secret` with anyone.

Access token	[redacted]
Access token secret	[redacted]
Access level	Read-only

Figure A.4 The OAuth settings

REFERENCES

[1] Mining Twitter Feeds for Top Stories CS 750 Data Mining Term Paper Kshitiz Bhattarai, George Mason University.

[2] Horn, Christopher. Analysis and Classification of Twitter messages. Diss. Master's thesis, Graz University of Technology, 2010.

[3] Abel, Fabian, et al. "Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web." Proceedings of the 3rd International Web Science Conference. ACM, 2011.

[4] Mazzia, Allie, and James Juett. "Suggesting hashtags on twitter." EECS 545m, Machine Learning, Computer Science and Engineering, University of Michigan (2009).

[5] Pennacchiotti, Marco, and Ana-Maria Popescu. "A machine learning approach to Twitter user classification." Fifth International AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[6] Kim, Younghoon, and Kyuseok Shim. "Twitobi: A recommendation system for Twitter using probabilistic modeling." Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE, 2011.

[7] Antenucci, Dolan, et al. "classification of tweets via clustering of hashtags." (2011).

[8] Chatterjee, Shaunak, Mobin Javed, and Anupam Prakash. "A time-sensitive user-specific recommendation system for Twitter."

[9] Hannon John, Mike Bennett, and Barry Smyth. "Recommending Twitter users to follow using content and collaborative filtering approaches."Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010.

[10] Kywe, Su Mon, et al. "On recommending hashtags in Twitter networks."Social Informatics. Springer Berlin Heidelberg, 2012. 337-350.

[11] Zangerle, Eva, Wolfgang Gassler, and Günther Specht. "Recommending#-tags in Twitter."Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings. Vol. 730. 2011.

[12] Pennacchiotti, Marco, and Siva Gurumurthy. "Investigating topic models for social media user recommendation."Proceedings of the 20th international conference companion on World wide web. ACM, 2011.

[13] Java, Akshay, et al. "Why we Twitter: understanding microblogging usage and communities."Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, 2007.

[14] Honey, Courtenay, and Susan C. Herring. "Beyond microblogging: Conversation and collaboration via Twitter." System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on. IEEE, 2009.

[15] Grosseck, Gabriela, and Carmen Holotescu. "Can we use Twitter for educational activities." 4th international scientific conference, eLearning and software for education, Bucharest, Romania. 2008

[16] Jansen, Bernard J., et al. "Micro-blogging as online word of mouth branding."CHI'09 Extended Abstracts on Human Factors in Computing Systems. ACM, 2009.

[17] Acar, Adam. "Culture and Social Media Usage: Analysis of Japanese Twitter Users." International Journal of Electronic Commerce Studies 4.1 (2013): 21-32.

[18] Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." Journal of Computational Science 2.1 (2011): 1-8.

[19] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of Machine Learning Research 3 (2003): 993-1022.

[20] Ramage, Daniel, Susan Dumais, and Dan Liebling. "Characterizing microblogs with topic models." International AAAI Conference on Weblogs and Social Media. Vol. 5. No. 4. 2010.

[21] Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-label corpora. EMNLP 2009

[22] Ramage, Daniel, Susan Dumais, and Dan Liebling. "Characterizing microblogs with topic models."International AAAI Conference on Weblogs and Social Media. Vol. 5. No. 4. 2010.

[23] M.Tech. Seminar Report Twitter Data Analysis Gaurish Chaudhari (113050037) gaurish@cse.iitb.ac.in April 12, 2012

[24] http://en.wikipedia.org/wiki/Twitter

[25] https://about.twitter.com/company

[26] Cheong, Marc. "‘What are you Tweeting about?’: A survey of Trending Topics within Twitter." Clayton School of Information Technology, Monash University(2009).

[27] Jansen, Bernard J., et al. "Micro-blogging as online word of mouth branding."CHI'09 Extended Abstracts on Human Factors in Computing Systems. ACM, 2009.

[28] https://dev.twitter.com/overview/documentation

[29] http://www.statista.com/chart/1520/number-of-monthly-active-twitter-users/

[30] www.twitter.com

[31] http://financialorbit.blogspot.com.tr/2014/11/twitter-my-ten-favourite-slides-from.html

[32] Akın, Ahmet Afsin, and Mehmet Dündar Akın. "Zemberek, an open source nlp framework for turkic languages." Structure 10 (2007).

[33] Sriram, Bharath, et al. "Short text classification in twitter to improve information filtering."Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2010.

[34] Huang, Anna. "Similarity measures for text document clustering." Proceedings of the sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand. 2008.

[35] Sorensen, Seth. "Accuracy of Similarity Measures in Recommender Systems." (2012).

[36] Leung, K. Ming. "Naive Bayesian Classifier." Polytechnic University Department of Computer Science/Finance and Risk Engineering (2007).

[37] Murphy, Kevin P. "Naive bayes classifiers." University of British Columbia(2006).

[38] MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.

[39] Wagstaff, Kiri, et al. "Constrained k-means clustering with background knowledge." ICML. Vol. 1. 2001.

[40] Kanungo, Tapas, et al. "An efficient k-means clustering algorithm: Analysis and implementation." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.7 (2002): 881-892.

[41] Jain, Anil K. "Data clustering: 50 years beyond K-means." Pattern recognition letters 31.8 (2010): 651-666.

[42] Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis."Discourse processes25.2-3 (1998): 259-284.

[43] Hu, Diane J. "Latent dirichlet allocation for text, images, and music." University of California, San Diego. Retrieved April 26 (2009): 2013.

[44] Canini, Kevin R., Lei Shi, and Thomas L. Griffiths. "Online inference of topics with latent Dirichlet allocation."Proceedings of the International Conference on Artificial Intelligence and Statistics. Vol. 5. No. 1999. 2009

[45] Walsh, Brian. "Markov chain monte carlo and gibbs sampling." (2004).

[46] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." Proceedings of the fifth annual workshop on Computational learning theory. ACM, 1992.

[47] Meyer, David, and FH Technikum Wien. "Support vector machines." The Interface to libsvm in package e1071 (2014).

[48] Hearst, Marti A., et al. "Support vector machines." Intelligent Systems and their Applications, IEEE 13.4 (1998): 18-28.

[49] Hannon, John, Mike Bennett, and Barry Smyth. "Recommending Twitter users to follow using content and collaborative filtering approaches."Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010.

[50] Chen, Kailong, et al. "Collaborative personalized tweet recommendation."Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012.

[51] Kywe, Su Mon, Ee-Peng Lim, and Feida Zhu. "A survey of recommender systems in Twitter."Social Informatics. Springer Berlin Heidelberg, 2012. 420-433.

[52] Van Meteren, Robin, and Maarten Van Someren. "Using content-based filtering for recommendation." Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop. 2000.

[53] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook." Recommender systems handbook. Springer US, 2011. 1-35.

[54] Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997).

[55] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. "Collaborative filtering recommender systems." Foundations and Trends in Human-Computer Interaction 4.2 (2011): 81-173

[56] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting web sites. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pages 54–61, Portland, OR, August 1996.

[57] Mooney, Raymond J., and Loriene Roy. "Content-based book recommending using learning for text categorization." Proceedings of the fifth ACM conference on Digital libraries. ACM, 2000.

[58] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." Recommender systems handbook. Springer US, 2011. 73-105.

[59] Lee, Joonseok, Mingxuan Sun, and Guy Lebanon. "A comparative study of collaborative filtering algorithms." arXiv preprint arXiv:1205.3193 (2012).

[60] Billsus, Daniel, and Michael J. Pazzani. "Learning Collaborative Information Filters."ICML. Vol. 98. 1998.

[61] Burke, Robin. "Hybrid recommender systems: Survey and experiments." User modeling and user-adapted interaction 12.4 (2002): 331-370.

[62] https://code.google.com/p/stop-words/

[63] Fox, Christopher. "A stop list for general text." ACM SIGIR Forum. Vol. 24. No. 1-2. ACM, 1989

[64] Porter, Martin F. "An algorithm for suffix stripping." Program 14.3 (1980): 130-137.

[65] Lovins, Julie B. Development of a stemming algorithm. MIT Information Processing Group, Electronic Systems Laboratory, 1968.