International Journal of Scientific Engineering and Science
Volume 1, Issue 12, pp. 8-14, 2017. ISSN (Online): 2456-7361

Discovering Similar Cities Using Text Mining: A Recommendation Application for Turkey

Yunus Doğan[1], Yunus Turdu[2]
[1, 2]Department of Computer Engineering, Dokuz Eylül University, Izmir, Turkey

***Abstract*— *The purpose of this study is to show that it is possible to benefit from the use of text mining to capture alternative cities that are targeted by a person on touristic journeys. The first process has been to collect the texts containing the descriptions of 100 cities and to convert them into a dataset form for text mining. Secondly, K-Means and the density-based spatial clustering of applications with noise (DBSCAN) algorithms have been used and compared to obtain similar cities. Multi-layer perceptron, Naïve-Bayes, K-Nearest Neighbor, Decision Tree and Support Vector Machine (SVM) algorithms have been used to classify these cities. Since Multi-Layer Perceptron yields over 70%, it has been determined to be the most successful algorithm for this purpose. The SOM (Self Organizing Map) algorithm has been used to obtain more consistent and accurate results of the distribution, and the clusters have been finalized. In analyzing of the application for Turkey, 28 cities in Turkey and 72 other cities have been evaluated and it has been possible to present the cities which have been similar to Turkey as alternatives. For this purpose, the obtained results from text mining have been visualized through a mobile application. The results of the analysis for the mobile application have been recorded in a database and presented to the user on the Android platform using the Windows Communication Foundation (WCF) web service methods.*

***Keywords*— *Text Mining, Natural Language Processing, Mobile Application, Web Service, Tourism.*

I. Introduction

Some city guide applications or websites use machine learning and data mining technique to present more correct results. For example, Triposo city guide application has sent advises as e-mail similar cities according to user's previous searches (triposo.com/travelguide). Also, Tripadvisor presents adversities according to user data and searches after logging into the websites (tripadvisor.com.tr/ Tourism-g293969-Turkey-Vacations.html). These applications have used user transaction data and analyzed by using of data mining techniques to make campaigns depend on these results.

Tourism may improve for by promotions giving by advisor websites. There are many applications for promotions of different cities. The main goal of our study is that finding most similar cities to the city selected by user with usage of text mining. For example, a user in Turkey wants to visit a city in Australia. When the user searches cities according to his/her inputs in this application, the system returns cities in Turkey as alternatives to the selected city in Australia. Thus, it can find similar cities in Turkey and the idea of the user can change not to go to a far city. For this purpose, the data of different cities have been collected; this data have been used and analyzed by using of text mining techniques. Finally, it has presented these results.

Firstly, different cities data have been collected from different websites (wikitravel.org and wikivoyage.org). After that, the operations of the natural language processing (NLP) have been implemented with a tool. It has been generated for data preparation. Trivial words, punctuation marks and stop-words have been removed in text files by the tool. These prepared data have been used for data mining techniques. The program has been decided the most similar cities according to the results of clustering algorithm. As Microsoft SQL has been used for data storage and user queries, Android SDK have been used for user interface and result representation. As a result, the obtained clusters contain other cities including the cities in Turkey, and then the application has been found the similar cities in Turkey to the selected city and showed the result set of cities in the mobile application.

Weka tool have used for data mining Clustering algorithm methodologies and the experimental studies and results have been explained in next sections in detail. Also, C# tools have used for data preparation. WCF web service, Microsoft .NET and Microsoft SQL have been used for entity, facade and presentation layer operations of the applications.

II. Related Works

Lorenzi et al. [1] have implemented a tourism recommender system based on textual analysis. Usually, travelers need some advices about where to go and what to do. The system use collected user data for text mining techniques. Some travel agencies use this system to help tourists. All words related to tourism called “Ontology” in the system, some words have been discarded in ontology (stop words, prepositions etc.). Travel agencies have created tourism categories. For example, adventure, tropical, beach, vacation, historical tourism. After that the customer and travel agent start answer and question in the system. Customers are classified by applying text mining methods to customer messages. The system suggests the customer according to these classes.

Lau et al. [2] have a study about text mining for the hotel industry. Business intelligence is important for hotel industry. Many studies propose text mining as a means of information management. There are many ways to collect data for text mining. Online collected data; different web pages, social media accounts, e-mails, news-groups are web-based data sources. The other way of collecting information is in guidebooks. The hotel-related datasets and words are collected in the database. Hotel profiles are analyzed based on text mining. There are features to consider when choosing a hotel (number of room, transportation, price, safety etc.). The main purpose of this project is to assist in hotel selection using text mining algorithms. This project was made for the hotels in Hong Kong.

In another study, Berezina et al. [3] have analyzed online hotel reviews by using text mining. The purpose of this article is to investigate satisfied and unsatisfied hotel customers. Therefore, 2,510 comments about Florida from “tripadvisor.com” website have been reviewed. Satisfied and unsatisfied customers have been compared. Some categories have identified for the positive review and the negative reviews recommendations. For example, a small distance between the beach and the hotel is a positive situation but it is a negative situation that the number of hotel employees is low. CATPAC, a text mining method of determining the frequency of words has been used in the project. Both positive and negative reviews have been important for text-mining process and they have been evaluated. For each word, they have been calculated the number of positive and negative reviews. The next process has been Text-Link. Text-Link analysis has identified new means of word groups.

The main purpose of the article of Segall et al. [4] is that SAS Text Miner and Megaputer Polyanalyst especially have applied for hotel consumer survey data. The increased textual knowledge has caused another increasing for text mining applications. There are a lot of text mining applications. This article has compared two selected applications according to hotel data. Hotels have been increased customer service quality by using text mining methods. These text mining tools has been applied in the different sectors.

Claster et al. [5] have used seventy-million tweets as dataset. Contains a tweet; user information, comment, location and date information. The article has been interested only three locations (Sri Lanka, Thailand and Mexico). Tweets have been filtered according include terms about tourism. SOM and Naive-Bayes algorithms have been used in the study. According to specific dates and locations tweets have been analyzed. For example; November and January months are summer in Sri Lanka. Therefore, the tweets in these months have been evaluated. Results of this study have been used in tourism sector.

Furthermore, it is possible to investigate some studies about tourism using data mining algorithms in literature (e.g. [6], [7] – [8]). For this aim, Dickinger et al. [9] have used SVM, Guo et al. [10] have used Sequential Pattern Mining Algorithms, Godnov et al. [11] have used Sentimental Analysis, Pembeci [12] has used Regression and Kernel-Density, Gassiot and Coromina [13] have used Multivariate statistical analyses and Claster et al. ([14] – [15]) have used Self-Organizing Maps, Naïve Bayes and unsupervised artificial neural networks.

III. METHODOLOGIES

The study has kept to the process of “knowledge discovery in database” (KDD). The first step has been for data selection, the second step has been for data pre-processing, the third step has been for data transformation, the forth step has been for data mining and the fifth step has been for pattern interpretation. In this section, the process of KDD is explained generally.

A. Data Selection

In this step, a data collecting operation has been done. For this reason, the sites that provide information about the cities have been searched and the data used in this study have been collected from Wikitravel.com and Wikivoyage.com websites.

B. Data Pre-processing

In this step, NLP operations have been done. The first one has been the elimination of punctuation marks. Punctuation marks may change the accuracy of the analysis. Such as “doing.” and “doing” words are not reduced into the same meaning in the analysis. Therefore, the punctuation marks have deleted to make to word-based processing.

The second operation has been the elimination of stop words. Some of the words in the sentence does not make any sense, like “a, an, the, of called stop words. These words would prevent to the analysis to get the right results. Therefore, the stop words have been removed.

The third operation has been the changing the words in irregular form into infinitive forms, because the meaning of sentence may change according to the tense of the English verb. For example; “write” word should change as “wrote” or “written” according to related time. All verbs have reduced to the infinitive forms. For these operations, a tool has been implemented in the study.

C. Data Transformation and Storage

In this study, firstly, the corpus has been collected in a database. Microsoft Sql (MSSQL) has been used as a database management system. Each city name and information has saved in this database. When users select any city, the system sends the most similar city information.

Secondly, a “.csv” file has been created for each city and the words about the city have been saved. The words about cities have saved a different “.csv” file and all files have been compared using some text-mining methodologies. The system has decided the most similar files according to result of text-mining algorithms.

D. Data Mining

At comparison phase, Weka tool has been used. The sub-functions have been different data mining algorithms such as; K-Nearest Neighbour, Multi-layer perceptron, Naïve Bayes, Decision Trees, Support Vector Machine (SVM), Self-Organizing Map (SOM), K-Means and DBSCAN.

E. Pattern Interpretation

In this phase, the outcomes from data mining algorithms have been evaluated and validated. The success rates of the algorithms have been compared and the pattern of the most successful algorithm has been assumed for the next phase of the mobile application.

The clusters with the cities have been stored in the fact database and WCF web service methods, which have used this database to connect the database with the mobile application, have been implemented.

Fig. 1 shows the architectural view of the system.

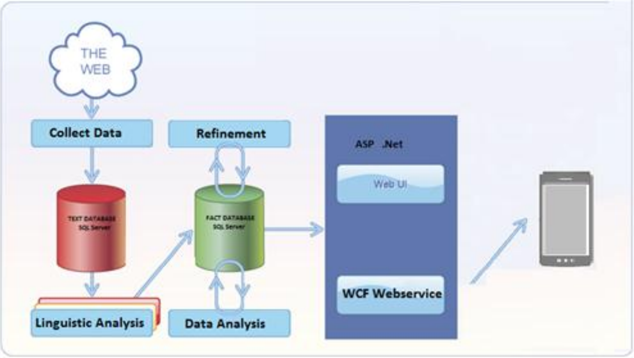


Fig. 1. The architectural view of the study.

IV. Experimental Studies

In order to compare the cities in the project and find similar ones, the articles to introduce these cities have been found. The different sources have been searched, but each source had to emphasize a different feature of cities. For example, while there is an introductory article concentrating on a rich Parisian cuisine, the other one is the fashion, culture and artistic features of the Parisian city. This could have caused the different results and illusions when comparing cities. Therefore, a resource had to offer all the features of the cities under certain headings and explain the different features. Furthermore, it had to contain all the city and tourist places in our database, it is possible to use “wikipedia.com” tourism related versions of “wikitravel.com” and “wikivoyage.com”. Finally, the related words have been collected and saved as ".txt" files.

For the selection of recorded cities, “telegraph.co.uk” has been selected because this site shows the best 50 cities in the world. (telegraph.co.uk/travel/galleries/The-worlds-best-cities-in-pictures/). Tourist attractions in Turkey have been also added to the storage. The designated tourist destinations are as shown in Table I and Table II.

TABLE I. Cities in Turkey taken part in the system.

Id	City	Id	City	Id	City	Id	City
1	Adana	8	Diyarbakır	15	Hatay	22	Konya
2	Ankara	9	Edirne	16	Istanbul	23	Marmaris
3	Antalya	10	Erzurum	17	Kapadokya	24	Samsun
4	Bodrum	11	Eskisehir	18	Kas	25	Tokat
5	Bursa	12	Fethiye	19	Kastamonu	26	Trabzon
6	Cesme	13	Gaziantep	20	Kayseri	27	Urfa
7	Denizli	14	Hasankeyf	21	Kemer	28	Van

After the data have been collected, the data preparation phase, which is one of the most important stages of the data mining, has been implemented. The words in English texts have been translated into plain words. At this stage in turn:

- Elimination of non-English words. Example: Sultan, Ahmet, Cami’s, they must be removed.
- Elimination of stop words. Example: The, an, a, at, is, etc.
- Elimination of comparative. Example: “bigger” changed “big” as base form.
- Elimination of superlative. Example: “biggest” changed “big” as base form.
- Elimination of suffixes. Example: “worker” changed “work” as base form.
- Transformation from past tense, past participle and present participle to infinitive forms. Example: “worked”,” working”,” works” changed “work” as base form.

The first process has been that deleting the punctuation marks in the texts. After that, the obtained words have been compared to the words in the English dictionary (www-01.sil.org/linguistics/wordlists/english/wordlist/wordsEn.txt).

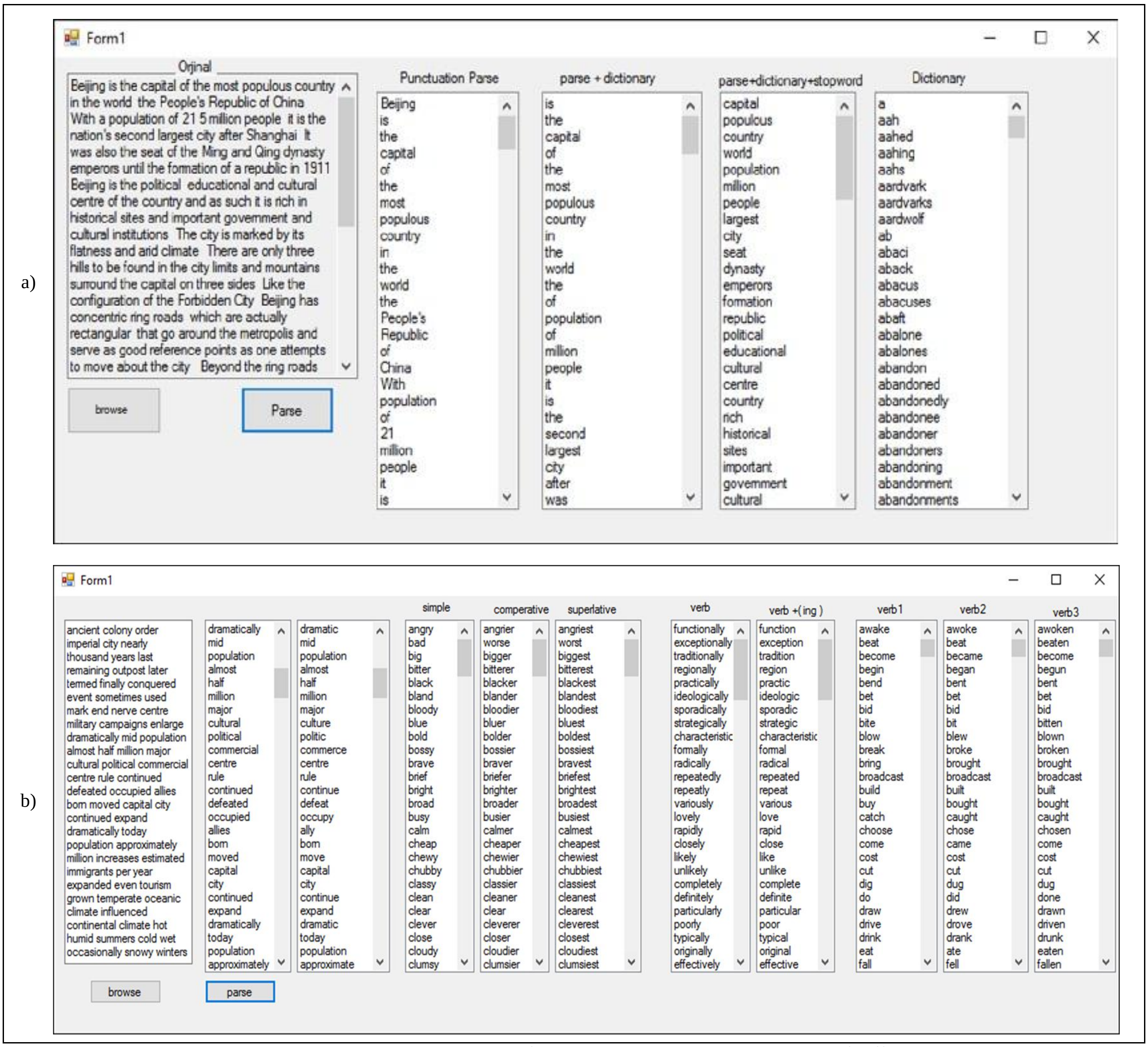
TABLE IIIII. Cities except Turkey taken part in the system.

Id	City	Id	City	Id	City	Id	City
1	Abu Dhabi	19	Chiang Mai	37	Lisbon	55	Rio de Jenario
2	Alaska	20	Cologne	38	London	56	San Francisco
3	Amsterdam	21	Copenhagen	39	Luxembourg	57	Santiago
4	Antarctica	22	Dakar	40	Lyon	58	Seoul
5	Baku	23	Dijon	41	Madrid	59	Seville
6	Bangkok	24	Dresden	42	Marrakech	60	Shanghai
7	Barcelona	25	Dubai	43	Milano	61	St Petersburg
8	Beijing	26	Dublin	44	Minsk	62	Stockholm
9	Belgrade	27	East Bourne	45	Monaco	63	Strasbourg
10	Berlin	28	Eindhoven	46	Moscow	64	Sydney
11	Birmingham	29	Florence	47	Munich	65	Tehran
12	Bochum	30	Hamburg	48	Nairobi	66	Toronto
13	Bordeaux	31	Helsinki	49	Napoli	67	Vancouver
14	Boston	32	Jakarta	50	New Delhi	68	Venice
15	Budapest	33	Kabul	51	New York	69	Vienna
16	Canberra	34	Krakow	52	Oslo	70	Warsaw
17	Cancun	35	Kuala Lumpur	53	Paris	71	Yerevan
18	Cape Town	36	Kyiv	54	Prague	72	Zagreb

The words, which do not exist in the dictionary, have been deleted from the data set. Furthermore, the stop-words in English, which mean that they do not contain any meaning if they do not take part in the sentences, have been deleted. The “a” part in Fig. 2 shows these operations.

In this phase, the text file we obtained has contained only English words which are not stop-words. Suffixes of adverbs, adjectives, superlatives, comparatives, past simple and past participle have been converted to the base forms of the words. These operations have been done as given in the “b” part in Fig. 2 and finally, the last version of the text file has been prepared for data mining operations.

After cleaning the data set, the files with the extension ".csv" have been needed for text mining. All city names had to be represented as rows and the cleaned words had to be represented as columns to storage specific frequencies in the data set. The next operation has been to discover the break point for the frequency in the data set, because the aim of data mining operations had to take the words which have counts higher than a certain frequency. The break point has been assumed as 2 for frequencies. Thus, the words as features, which have more frequency than 2, would be able to be used to compare 2 cities.



Orjinal

Beijing is the capital of the most populous country in the world the People's Republic of China With a population of 21 5 million people it is the nation's second largest city after Shanghai It was also the seat of the Ming and Qing dynasty emperors until the formation of a republic in 1911 Beijing is the political educational and cultural centre of the country and as such it is rich in historical sites and important govemment and cultural institutions The city is marked by its flatness and arid climate There are only three hills to be found in the city limits and mountains surround the capital on three sides Like the configuration of the Forbidden City Beijing has concentric ring roads which are actually rectangular that go around the metropolis and serve as good reference points as one attempts to move about the city Beyond the ring roads

Punctuation Parse	parse + dictionary	parse+dictionary+stopword	Dictionary
Beijing	is	capital	a
is	the	populous	aah
the	capital	country	aahed
capital	of	world	aahing
of	the	population	aahs
the	most	million	aardvark
most	populous	people	aardvarks
populous	country	largest	aardwolf
country	in	city	ab
in	the	seat	abaci
the	world	dynasty	aback
world	the	emperors	abacus
the	of	formation	abacuses
People's	population	republic	abaft
Republic	of	political	abalone
of	million	educational	abalones
China	people	cultural	abandon
With	it	centre	abandoned
population	is	country	abandonedly
of	the	rich	abandonee
21	second	historical	abandoner
million	largest	sites	abandoners
people	city	important	abandoning
it	after	govemment	abandonment
is	was	cultural	abandonments

			simple	comperative	superlative	verb	verb +(ing)	verb1	verb2	verb3
ancient colony order	dramatically	dramatic	angry	angrier	angriest	functionally	function	awake	awoke	awoken
imperial city nearly	mid	mid	bad	worse	worst	exceptionally	exception	beat	beat	beaten
thousand years last	population	population	big	bigger	biggest	traditionally	tradition	become	became	become
remaining outpost later	almost	almost	bitter	bitterer	bitterest	regionally	region	begin	began	begun
termed finally conquered	half	half	black	blacker	blackest	practically	practic	bend	bent	bent
event sometimes used	million	million	bland	blander	blandest	ideologically	ideologic	bet	bet	bet
mark end nerve centre	major	major	bloody	bloodier	bloodiest	sporadically	sporadic	bid	bid	bid
military campaigns enlarge	cultural	culture	blue	bluer	bluest	strategically	strategic	bite	bit	bitten
dramatically mid population	political	politic	bold	bolder	boldest	characteristic	characteristic	blow	blew	blown
almost half million major	commercial	commerce	bossy	bossier	bossiest	formally	formal	break	broke	broken
cultural political commercial	centre	centre	brave	braver	bravest	radically	radical	bring	brought	brought
centre rule continued	rule	rule	brief	briefer	briefest	repeatedly	repeated	broadcast	broadcast	broadcast
defeated occupied allies	continued	continue	bright	brighter	brightest	repeatly	repeat	build	built	built
bom moved capital city	defeated	defeat	broad	broader	broadest	variously	various	buy	bought	bought
continued expand	occupied	occupy	busy	busier	busiest	lovely	love	catch	caught	caught
dramatically today	allies	ally	calm	calmer	calmest	rapidly	rapid	choose	chose	chosen
population approximately	bom	bom	cheap	cheaper	cheapest	closely	close	come	came	come
million increases estimated	moved	move	chewy	chewier	chewiest	likely	like	cost	cost	cost
immigrants per year	capital	capital	chubby	chubbier	chubbiest	unlikely	unlike	cut	cut	cut
expanded even tourism	city	city	classy	classier	classiest	completely	complete	dig	dug	dug
grown temperate oceanic	continued	continue	clean	cleaner	cleanest	definitely	definite	do	did	done
climate influenced	expand	expand	clear	clear	clearest	particularly	particular	draw	drew	drawn
continental climate hot	dramatically	dramatic	clever	cleverer	cleverest	poorly	poor	drive	drove	driven
humid summers cold wet	today	today	close	closer	closest	typically	typical	drink	drank	drunk
occasionally snowy winters	population	population	cloudy	cloudier	cloudiest	originally	original	eat	ate	eaten
	approximately	approximate	clumsy	clumsier	clumsiest	effectively	effective	fall	fell	fallen

Fig. 2. a) The parsing and elimination of non-English words. b) The elimination of suffixes.

In the next phase, the operations for the creation of ".csv" files, which have the words with different break points for the frequency, have been performed.

TABLE IVVVI. The standard deviation values by frequency.

Freq.	K = 15	K = 10	K = 7	K = 5	K = 3
2	20.31	26.38	18.29	27.89	34.12
5	14.06	19.37	28.12	40.82	52.57
10	16.81	21.60	18.48	26.98	40.67
12	13.10	18.79	19.48	29.61	39.95
15	9.78	12.46	15.22	17.01	28.36
18	7.51	10.78	12.09	7.14	24.00
20	10.60	15.04	16.72	16.83	36.25
30	6.97	8.39	12.80	18.93	36.17
35	4.67	7.42	11.75	16.37	34.58
40	4.80	7.10	10.51	11.61	17.09
45	3.75	6.56	11.51	13.28	17.92
50	4.70	7.11	11.64	14.35	18.23
55	4.70	8.51	8.63	12.88	26.27
60	5.15	7.03	9.06	8.15	14.18

It has been to record the words higher frequency than a certain one after obtaining the general frequencies. For example, the files with frequencies of 60, 55, 50, 45, …, 10, and 5 have been created for words separately. Finally, the words with a certain frequency have been saved as separate files with ".csv" extensions.

A common file consisting frequency information for each city has been created as given in Fig. 3. In this sample shows the frequencies of each word for each city as a matrix.

Respectively, the standard deviations of each file for the frequency numbers of 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, and 60 after obtaining files using the Weka tool have been calculated. Table III shows that the standard deviation value is the minimum when the frequency is 45. Also, this result means the maximum consistency. After this step, "frequency45.csv" file have been obtained. The data set has been divided into certain k-valued clusters by K-Means method using Weka tool. When the standard deviation values of this file have been compared with k number of 3, 5, 7, 10 and15, it has been seen that the most consistent value has been obtained for 15. This observation has pointed out that the ideal distribution would be obtained when the cities had to be divided into 15 clusters.

After the "frequency45.csv" file has been divided into 15 clusters using the K-Means algorithm, the number of elements for each cluster is as shown in the first column in Table V. The centres of these clusters have been obtained with the Weka tool as given in Fig. 4. The Self Organizing Map (SOM) algorithm has been used to obtain a better distribution of the clusters, because the 5th cluster had only one city.

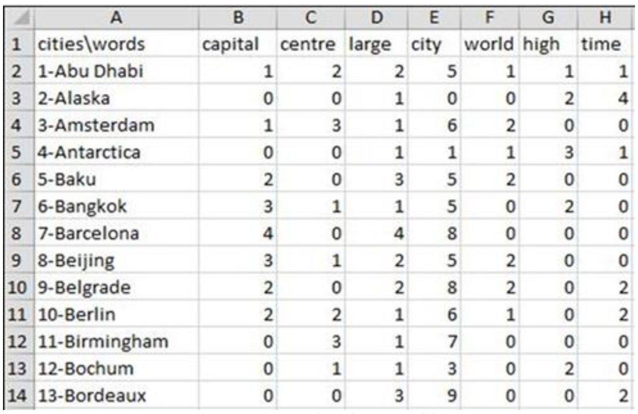
SOM algorithm has been evaluated to discover the closest neighbour to merge the cluster with its closest neighbour. In this study, a recommendation mechanism has been the aim; therefore, all clusters had to have cities at least two. After SOM, the coordinates in Table IV have been obtained.

TABLE VIIIV. Coordinates of clusters according to SOM algorithm.

Id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
X	0	0	2	3	0	0	3	2	0	0	3	1	0	1	0
Y	0	0	2	1	2	0	3	1	0	0	0	1	0	3	3

The 5th cluster with one element had to be included into one of the other cluster in the same coordinate. When the distances between the centres of the clusters have been measured according to the Euclidean distance, it has been seen that the closest cluster centre has been the centre of cluster 0. For this reason, the single element in the cluster 5 has transferred into the cluster 0.

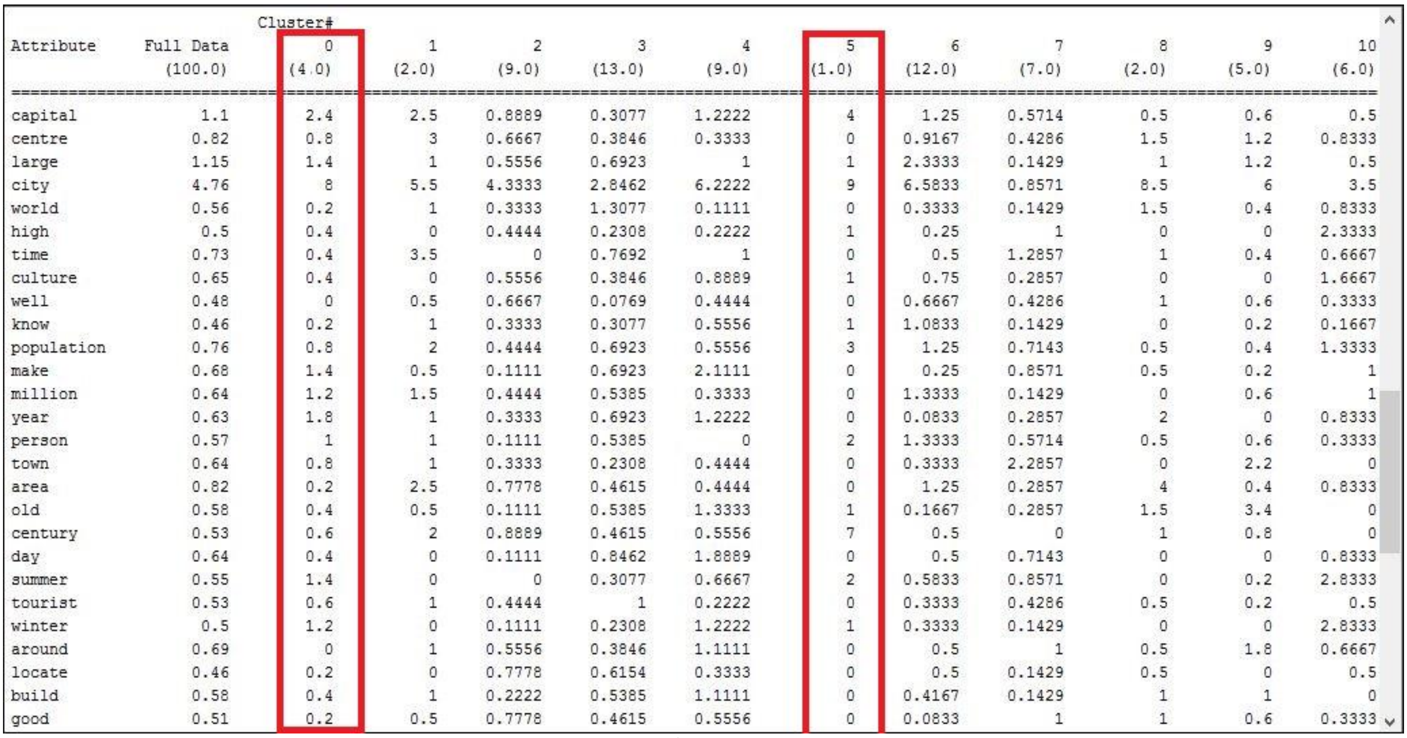
The number of elements in the clusters after K-Means and SOM operations and their differences are shown in Table V. Before applying the SOM algorithm, the standard deviation value had been 3.75, which have been 2.70 after the SOM algorithm has been applied. This shows that after the implementation of the SOM algorithm, the number of elements in the clusters has more consistent and better distribution.



	A	B	C	D	E	F	G	H
1	cities\words	capital	centre	large	city	world	high	time
2	1-Abu Dhabi	1	2	2	5	1	1	1
3	2-Alaska	0	0	1	0	0	2	4
4	3-Amsterdam	1	3	1	6	2	0	0
5	4-Antarctica	0	0	1	1	1	3	1
6	5-Baku	2	0	3	5	2	0	0
7	6-Bangkok	3	1	1	5	0	2	0
8	7-Barcelona	4	0	4	8	0	0	0
9	8-Beijing	3	1	2	5	2	0	0
10	9-Belgrade	2	0	2	8	2	0	2
11	10-Berlin	2	2	1	6	1	0	2
12	11-Birmingham	0	3	1	7	0	0	0
13	12-Bochum	0	1	1	3	0	2	0
14	13-Bordeaux	0	0	3	9	0	0	2

Fig. 3. A sample of ".csv" file.

Finally, the calculation of the standard deviations of words has been done to find out which words in each cluster had to be used. According to the result of the SOM algorithm, the frequencies of the 0th and 5th clusters have been needed to combine by a written code to calculate the weighted average of the centroids. As a result, a decreasing has been observed for the number of clusters from 15 to 14. In the next sections, the precision, recall and f-measure values are evaluated for the numbers of words to obtain the optimum break point according to the classification algorithms.



```
                                  Cluster#
Attribute        Full Data           0          1          2          3          4          5          6          7          8          9         10
                   (100.0)       (4.0)      (2.0)      (9.0)     (13.0)      (9.0)      (1.0)     (12.0)      (7.0)      (2.0)      (5.0)      (6.0)
=====================================================================================================================================================
capital                1.1         2.4        2.5     0.8889     0.3077     1.2222          4       1.25     0.5714        0.5        0.6        0.5
centre                0.82         0.8          3     0.6667     0.3846     0.3333          0     0.9167     0.4286        1.5        1.2     0.8333
large                 1.15         1.4          1     0.5556     0.6923          1          1     2.3333     0.1429          1        1.2        0.5
city                  4.76           8        5.5     4.3333     2.8462     6.2222          9     6.5833     0.8571        8.5          6        3.5
world                 0.56         0.2          1     0.3333     1.3077     0.1111          0     0.3333     0.1429        1.5        0.4     0.8333
high                   0.5         0.4          0     0.4444     0.2308     0.2222          1       0.25          1          0          0     2.3333
time                  0.73         0.4        3.5          0     0.7692          1          0        0.5     1.2857          1        0.4     0.6667
culture               0.65         0.4          0     0.5556     0.3846     0.8889          1       0.75     0.2857          0          0     1.6667
well                  0.48           0        0.5     0.6667     0.0769     0.4444          0     0.6667     0.4286          1        0.6     0.3333
know                  0.46         0.2          1     0.3333     0.3077     0.5556          1     1.0833     0.1429          0        0.2     0.1667
population            0.76         0.8          2     0.4444     0.6923     0.5556          3       1.25     0.7143        0.5        0.4     1.3333
make                  0.68         1.4        0.5     0.1111     0.6923     2.1111          0       0.25     0.8571        0.5        0.2          1
million               0.64         1.2        1.5     0.4444     0.5385     0.3333          0     1.3333     0.1429          0        0.6          1
year                  0.63         1.8          1     0.3333     0.6923     1.2222          0     0.0833     0.2857          2          0     0.8333
person                0.57           1          1     0.1111     0.5385          0          2     1.3333     0.5714        0.5        0.6     0.3333
town                  0.64         0.8          1     0.3333     0.2308     0.4444          0     0.3333     2.2857          0        2.2          0
area                  0.82         0.2        2.5     0.7778     0.4615     0.4444          0       1.25     0.2857          4        0.4     0.8333
old                   0.58         0.4        0.5     0.1111     0.5385     1.3333          1     0.1667     0.2857        1.5        3.4          0
century               0.53         0.6          2     0.8889     0.4615     0.5556          7        0.5          0          1        0.8          0
day                   0.64         0.4          0     0.1111     0.8462     1.8889          0        0.5     0.7143          0          0     0.8333
summer                0.55         1.4          0          0     0.3077     0.6667          2     0.5833     0.8571          0        0.2     2.8333
tourist               0.53         0.6          1     0.4444          1     0.2222          0     0.3333     0.4286        0.5        0.2        0.5
winter                 0.5         1.2          0     0.1111     0.2308     1.2222          1     0.3333     0.1429          0          0     2.8333
around                0.69           0          1     0.5556     0.3846     1.1111          0        0.5          1        0.5        1.8     0.6667
locate                0.46         0.2          0     0.7778     0.6154     0.3333          0        0.5     0.1429        0.5          0        0.5
build                 0.58         0.4          1     0.2222     0.5385     1.1111          0     0.4167     0.1429          1          1          0
good                  0.51         0.2        0.5     0.7778     0.4615     0.5556          0     0.0833          1          1        0.6     0.3333
```

Fig. 4. The centroids before SOM implementation.

TABLE V. The number of elements in the clusters after K-Means and SOM

Cluster Name	After K-Means, the number of elements	After SOM, the number of elements
Cluster 0	4	5
Cluster 1	2	2
Cluster 2	8	8
Cluster 3	9	9
Cluster 4	8	8
Cluster 5	1	-
Cluster 6	10	10
Cluster 7	7	7
Cluster 8	2	2
Cluster 9	5	5
Cluster 10	6	6
Cluster 11	8	8
Cluster 12	6	6
Cluster 13	16	16
Cluster 14	8	8
Total	100	100
Std. Deviation	3.75	2.86

V. Analysis Results

After the process of the clustering, the pattern has been tested using classification algorithms. The cluster value for each city has been used as the target attribute in the data set and new data set files have been created for this analysis. To extend the confidence bounds, it has been assumed that as the obtained cluster values had to be stable, the words as the attributes in the data set had to be changed. Therefore, the break points according to the frequencies have changed and 8 ".csv" files, which have contained 3, 5, 7, 10, 15, 20, 25 and 30 words and have had a harmonious structure with a target attribute for the classification algorithms, have been created.

Subsequently, all files have been classified with different classification techniques. The used classification algorithms have been as follows; Support Vector Machines (SMO), Multilayer Perceptron, Naïve Bayes, K-Nearest Neighbour (IBk), Decision Tree (J48)

These algorithms have been compared using f-measure in (1), precision in (2) and recall in (3) terms. In Weka tool, these terms are named as follows; Weighted Average Precision, Weighted Average Recall and Weighted Average F-measure.

$$F-measure = 2\,\frac{Precision.Recall}{Precision+Recall}$$

$$Precision = \frac{|\{revelant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

$$Recall = \frac{|\{revelant\ documents\} \cap \{retrieved\ documents\}|}{|\{revelant\ \ documents\}|}$$

In Table VI, the precision, the recall and the f-measure values are listed for SMO algorithm. In Table VII, the values for the same terms are listed for Multilayer Perceptron. In Table VIII, the values are listed for Naïve Bayes.

In Table IX, the values are given for SMO algorithm and the values are listed for J48 algorithm in Table X. All values have been obtained separately for each file. For example, the first file has 100 instances and 4 attributes with 3 words as the features and a target attribute.

TABLE VI. Precision, Recall, F-measure values for SMO algorithm.

Number of Words	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F-Measure
3 words	0.067	0.200	0.094
5 words	0.201	0.300	0.209
7 words	0.374	0.380	0.317
10 words	0.513	0.440	0.393
15 words	0.575	0.510	0.494
20 words	0.622	0.620	0.595
25 words	0.655	0.650	0.630
30 words	0.654	0.660	0.637

TABLE VII. Precision, Recall, F-measure values for Multilayer Perceptron.

Number of Words	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F-Measure
3 words	0.067	0.200	0.094
5 words	0.253	0.290	0.268
7 words	0.271	0.290	0.278
10 words	0.444	0.430	0.432
15 words	0.563	0.540	0.542
20 words	0.654	0.650	0.642
25 words	0.712	0.690	0.689
30 words	0.643	0.640	0.628

TABLE VIII. Precision, Recall, F-measure values for Naïve Bayes.

Number of Words	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F-Measure
3 words	0.206	0.260	0.226
5 words	0.315	0.330	0.315
7 words	0.365	0.290	0.311
10 words	0.410	0.390	0.380
15 words	0.411	0.400	0.382
20 words	0.401	0.430	0.405
25 words	0.407	0.440	0.415
30 words	0.412	0.440	0.414

TABLE IX. Precision, Recall, F-measure values for IBk algorithm.

Number of Words	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F-Measure
3 words	0.136	0.200	0.161
5 words	0.136	0.200	0.161
7 words	0.320	0.280	0.281
10 words	0.416	0.370	0.366
15 words	0.389	0.420	0.380
20 words	0.545	0.530	0.504
25 words	0.468	0.480	0.450
30 words	0.467	0.480	0.452

TABLE X. Precision, Recall, F-measure values for J48 algorithm.

Number of Words	Weighted Avg. Precision	Weighted Avg. Recall	Weighted Avg. F-Measure
3 words	0.142	0.190	0.158
5 words	0.187	0.220	0.199
7 words	0.197	0.220	0.206
10 words	0.274	0.280	0.276
15 words	0.255	0.290	0.268
20 words	0.400	0.400	0.392
25 words	0.381	0.370	0.367
30 words	0.368	0.350	0.349

According to these results, it has been observed that the f-measure values of the Multilayer Perceptron technique has been obtained more successfully than the other techniques in the previous 5 tables. Also, it has been said that the file with 25 words has had the optimum number of the attributes.

VI. Mobile Application

The mobile application has been implemented on Android 5.0. This application contains 4 main operations; finding the most similar cities to a city, finding the most similar Turkish cities to a city, finding the cities that contain the selected word in the introduction, and finding the Turkish cities that contain the selected word in the introduction.

Fig. 5 shows two screenshots of the application. The first screenshot contains the menu of the operations and the second one presents the cities list to discover alternative cities.

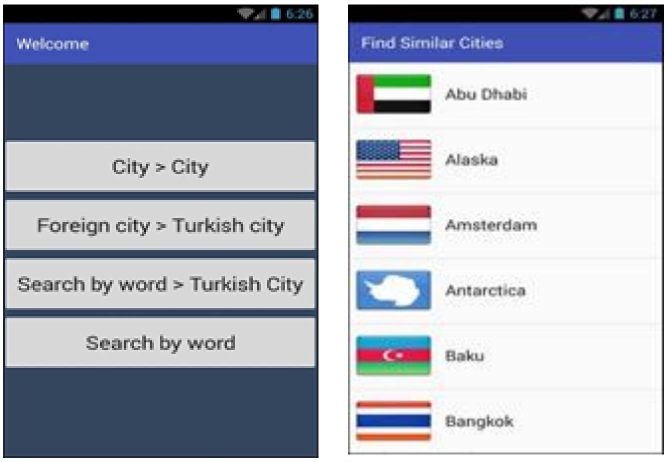


Fig. 5. The screenshots of the mobile application

Fig. 5 shows two screenshots of the application. The first screenshot contains the menu of the operations and the second one presents the cities list to discover alternative cities. This mobile application has been implemented to test the results of text mining and obtain the visual information.

VII. Conclusion

The study has been developed to find similarities among a certain number of cities using various text mining methods and present them to the user. In the study, 100 cities have been processed and this number should be increased in the next studies. The city information containing 16820 words has been collected from the different sources and they have been saved as files. NLP operations have been implemented on these files and clean data sets have been obtained for text mining methods. This study has been applied for a total of 100 cities, 27 of which have been Turkish cities and 73 have been the cities in the other countries. The cities have been divided into specific clusters using clustering algorithms. The SOM algorithm has been used to make the distribution better and to gain the balance of the distribution. Finally, 100 cities have been located in 14 clusters. Among the classification methods, Multilayer Perceptron had more consistent results with averagely 70% for the precision, the recall and the f-measure values than other methods. It shows that a new city with the frequency values for the certain 25 words can be located into a cluster which is discovered by Multilayer Perceptron.

In this study, the obtained results have been saved in MSSQL database and presented to the user in mobile environment using WCF web service. The main consequence of this study is to show that implementing text mining algorithms on web texts about the tourism sector can bring out the efficient applications containing mixing of an academic background and commercial software tools.

References

[1] F. Lorenzi, R. Saldana, S. Loh and D. Litchnow, “A Tourism Recommender System Based on Collaboration and Text Analysis”, *Information Technology & Tourism*, vol. 6(3), pp. 157-165, 2003.

[2] K. Lau, K. Lee and Y. Ho, “Text Mining for the Hotel Industry”, *Cornell Hotel and Restaurant Administration Quarterly*, vol. 46(3), pp. 344-362, 2005.

[3] K. Berezina, F. Okumus, A. Bilgihan and C. Cobanoglu, “Understanding Satisfied and Dissatisfied Hotel Customers: Text Mining of Online Hotel Reviews”, *Journal of Hospitality Marketing & Management*, vol. 25(1), pp. 1-24, 2016.

[4] R. Segall, Q. Zhang, and M. Cao, "Web-Based Text Mining of Hotel Customer Comments Using SAS Text Miner and Megaputer Polyanalyst", *SWDSI 2009 Proceedings*, pp. 141-152, 2009.

[5] W. Claster, W., M. Cooper, M., K. Tajeddini and P. Pardo, “Tourism, travel and tweets: algorithmic text analysis methodologies in tourism”, *Middle East J. Management*, vol. 1(1), pp. 81-99, 2013

[6] W. J. Amadio and J. D. Procaccino, "Competitive analysis of online reviews using exploratory text mining", *Tourism and Hospitality Management*, vol. 22(2), pp. 193-210, 2016.

[7] Y. Yifan, D. Junping, F. Dan and J. Lee, "Design and implementation of tourism activity recognition and discovery system", *in Proc. Intelligent Control and Automation (WCICA)*, Guilin, China, pp. 781-786, 2016

[8] H. Ban, H. Kimura and T. Oyabu, "Feature extraction of English guidebooks for Hokuriku region in Japan", *Journal of Global Tourism Research*, vol. 1(1), pp. 71-76, 2016.

[9] A. Dickinger, D. Astrid, L. Lidija, M. Josef and M. Josef, "Exploring the generalizability of discriminant word items and latent topics in online tourist reviews", *International Journal of Contemporary Hospitality Management*, vol. 29(2), pp. 803-816, 2017.

[10] T. Guo, B. Guo, Y. Ouyang, Z. Yu, J. C. Lam and V. O. Li, "CrowdTravel: scenic spot profiling by using heterogeneous crowdsourced data", *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-10, 2017.

[11] U. Godnov and T. Redek, "Application of text mining in tourism: Case of Croatia", *Annals of Tourism Research*, vol. 58, pp. 162-166. 2016.

[12] I. Pembeci, "Using Word Embeddings for Ontology Enrichment", *International Journal of Intelligent Systems and Applications in Engineering*, vol. 4(3), pp.49-56, 2016.

[13] A. Gassiot and L. Coromina, "Destination image of Girona: an online text-mining approach", *International Journal of Management Cases*, vol. 15(4), pp. 301-314, 2013.

[14] W. B. Claster, H. Dinh and M. Cooper, "Naïve Bayes and unsupervised artificial neural nets for Cancun tourism social media data analysis", *In Nature and Biologically Inspired Computing (NaBIC)*, Kitakyushu, Japan, pp. 158-163, 2010.

[15] W. B. Claster, M. Cooper and P. Sallis, "Thailand--Tourism and conflict: Modeling sentiment from Twitter tweets using naïve Bayes and unsupervised artificial neural nets", *in Proc. Computational Intelligence, Modelling and Simulation (CIMSiM)*, Tuban, Indonesia, pp. 89-94, 2010.