VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

Benchmark Effect of Web Search Engines on Text Mining

Ahmet TOPRAK[1]*, Metin TURAN[2]

[1]*Istanbul Commerce University, Computer Engineering, Istanbul*
[2]*Istanbul Commerce University, Computer Engineering, Istanbul*

Makale Bilgisi

Başvuru:
21/07/2020
Kabul:
15/11/2020

Abstract

There have been many studies about creating a dictionary and these studies have come from past to present with different methods and different analyzes. Especially with the emergence of the World Wide Web, efforts to create dictionary based on instant data have gained importance. Therefore, the performance of the web search engines directly effects the model which is using web documents for automatic dictionary creation. The web search engines were evaluated in terms of their suggested documents relationality to the query in the research. For this purpose, an automatic dictionary creating model using web documents were developed. First of all, the topic seed words are determined by the documents presented to the system initially. Search is executed by these seed words initially. Then TF-IDF metric was used as meaningful word selection method for returned first document. The top n meaningful words were selected from the highest TF-IDF values. The value of n was determined experimentally. When searching the web with these words added to the dictionary, new documents were suggesting by the web search engine. By repeating the process, experimental dictionaries of a certain size were obtained. By the way, the documents suggested by each web engine are generally different, so that the dictionary similarity produced from the top suggested documents can measure web engines performance of selecting relational documents. Hash similarity was used to evaluate dictionary performance. According to the results, dictionary with the 73.9% highest similarity for Google search engine, dictionary with the 68.7% highest similarity for Bing search engine and dictionary with the 60.5% highest similarity for Yandex search engine were produced.

Keywords: *Automatic Dictionary Creation, Hash Similarity, Natural Language Processing, Performance of Web, TF-IDF Metric*

* İletişim e-posta: ahmetoprak190363@gmail.com
** Part of this work presented orally at the III. International Conference on Data Science and Applications 2020.

1 Introduction

Nowadays the amount of data has increased considerably. If a search is required, then web search engine is the main device to achieve this goal nowadays. They search documents on the web using some simple rules and keywords given. The documents suggested by web search engines are generally looking for keywords match in the document without checking the content this document which is related or not. Within this study, in order to create automatic language dictionary about a topic using well-known different web search engines were experimented in terms of relationality of their suggested documents. The motivation of the idea is that if the documents suggested by each web engine are generally different, so that the dictionary similarity produced from the top suggested documents can measure web engines performance of selecting relational documents.

Dictionary creation studies include different techniques. Previous studies are usually hand-made [1], and require human intervention [2]. Today's dictionary studies are generally designed to be semi-automatic [3], or automatic [4]. It is observed that the results change in good direction depending on these techniques. The purpose of these studies is to obtain the desired information using the piles of documents.

In order to see the effects of web search engines on automatic dictionary creation, so as to test their success in presenting meaningful documents with top priority. For this purpose, the seed word (s) or document (s) given by the user as a reference are based on the system and automatic language dictionaries specific to the subject have been created [5]. In the study, 3 different search engine APIs, Google, Bing and Yandex were used for web search. With the documents obtained from these search engine APIs, the system was fed again and new dictionary words were obtained.

In the second section of the article, previous dictionary studies in literature or studies covering methods applied in this study are mentioned. In the third section, the methods applied in the automatic dictionary study are explained in detail. In the fourth section, the dictionaries and performance ratios obtained with the input parameters are given. In the last section, the results obtained and future studies are discussed.

2 Related Works

There have been many studies about creating a dictionary and these studies have come from past to present with different methods and different analyzes. Firstly, these studies were started in 1990 [2], and it is seen that achievements which can be obtained better results from such studies have been achieved in time. Especially with the emergence of the World Wide Web, efforts to create a dictionary based on instant data have gained momentum.

TF-IDF metric plays an important role in keyword selection. One of these studies is the work of Christian Caldera and colleagues [6]. In this study, a tool called PRIMA was developed. The purpose of this tool is to assign applications to the members of the International Program Committee (IPM), which causes the most workload at conferences. Because of each referee must be assigned an article according to his / her field of expertise. For this purpose, the articles uploaded to the system were examined automatically and the assignment process was carried out according to the field of expertise of the referees. TF-IDF metric was used to assign the appropriate arbitrator. In the experiments, it took 3 minutes to appoint arbitrators for 100 articles.

Another successful study of TF-IDF metric is the study of Apra Mishra et al [7]. In this study, the data set of FIRE 2011 was used. The aim of the study is to analyze the retrieval feature of the vector space model and create the most effective model. TF-IDF metrics were used for keyword detection in the study. The open search engine Terrier 3.5 was used for all experiments and evaluations. The results show that the TF-IDF metric gives the highest precision values with the new corpus dataset.

As stated in the previous section, the dictionary creation process is operated in three different ways; automatic, semi-automatic and manual. In 2015, Julia Lavid and her colleagues [8], created an automatic dictionary called MULTINOT, which contains concepts specific to the English-Spanish language. The MULTINOT corpus consists of original and translated texts and is designed as a multifunctional resource for use in a range of disciplines, such as lexical antonyms. The main purpose of the MULTINOT corpus is to create a parallel English-Spanish corpus that is balanced in terms of recording diversity and translation aspects and whose design and enrichment with linguistic explanations focuses on quality rather than quantity.

In 2015, Asif Iqbal Sarkar with his team [9], worked on automatic dictionary creation. In this work addressed the issue of automatic Bangla corpus creation, which will significantly help the processes of lexicon development, morphological analysis, automatic parts of speech detection and automatic grammar extraction and machine translation. The aim was to collect all free Bangla documents on the World Wide Web and offline documents available and extract all the words in them to make a huge repository of text.

In 2016, Beata Megyesi and sharers [10], created a manual automatic language dictionary called Uppsala. The Uppsala corpus consists of Swedish texts that different groups of students studying in Swedish language between the ages of nine and nineteen encounter in the country exam. A compilation has been created from texts belonging to all national exams conducted since 1996. This collection consists of 2,500 texts with more than 1.5 million symbols.

3 Current Experimental System

The experimental system developed for automatic dictionary creation is a product of MSc Thesis [5]. It takes seed document/s in order to determine dictionary subject. The document/s presented to the system must be subjected to the preprocessing steps. After the pre-processing steps, it is ensured that the meaningful words are determined. These words are used to feed the dictionary and the following web search. The current general system schema is given by Figure 1.

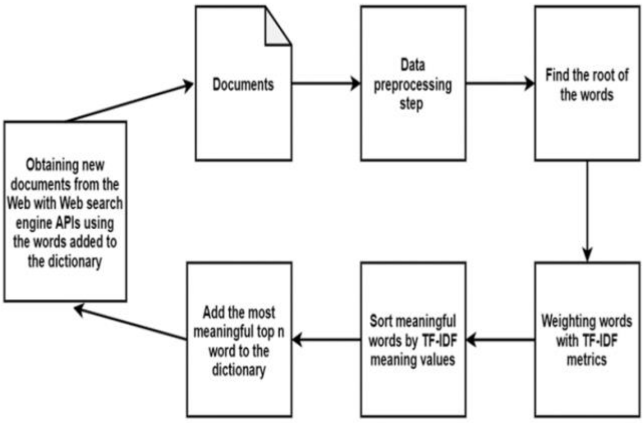

Documents
Data preprocessing step
Find the root of the words
Weighting words with TF-IDF metrics
Sort meaningful words by TF-IDF meaning values
Add the most meaningful top n word to the dictionary
Obtaining new documents from the Web with Web search engine APIs using the words added to the dictionary

Fig. 1. General structure of automatic dictionary creation

3.1 Pre-processing of documents

The most important point of achieving successful results in text mining studies is that the data to be used should be of high quality. In order to obtain quality data, the data must be pre-processed. In many applications, more than one of the types of data preprocessing is needed. Therefore, it is important to determine the type in data preprocessing [11]. Based on this approach, a large number of data preprocessing techniques have been developed. They are data cleaning, data consolidation, data conversion and discretization, data reduction techniques. The diagram below is used to depict the various steps involved in data preprocessing [12].

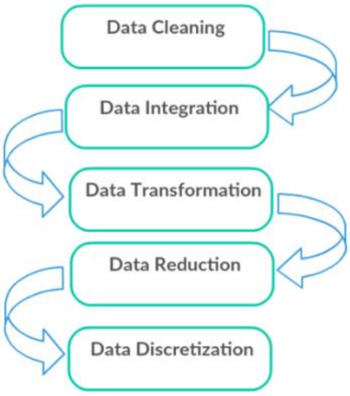


Fig. 2. Data preprocessing techniques

In the study, the documents collected from various sources are pre-processed. Document/s types and formats can be supplied differently. This requires pre-processing of the data with the data cleaning method to bring the documents to the appropriate format. As a first step, the document/s entered into the system should be separated into the units where they can be processed. In the current study, the procedures are considered as paragraph-based. For this reason, the fields of the documents with the <p> tag should be determined by separating them into HTML tags. The Beautiful Soup HTML parser library is used to separate documents into HTML tags. Beautiful Soup is a powerful and fast library for processing HTML or XML files. Thanks to this library, fields with the <p> tag are detected. Snippets of text separated into paragraphs are separated into words on a paragraph basis. Documents contain many words in the text that have no meaning, and when these words are removed from the sentence, they do not cause a semantic loss. However, if these words are not pre-processed, they will have a negative effect on the results. Because it will affect the results since it will be performed on a paragraph basis and meaningful words in this paragraph will be determined on frequency basis. Since the English dictionary will be created in the study, the end words of the English language should be removed from the documents. Below are the commonly used terminology words in English. "a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are", "aren't" etc. Since the stop words do not have a distinguishing feature, they are cleared during preprocessing.

In the next step, meaningless words such as spaces and numbers are removed from the document. Then all the words in the document are written in lower case format to ensure letter compatibility. Later, the construction of the words and affixes are converted into a single form. The reason for this process is to find meaningful words at the stage of frequency-based processing is done. The stemming algorithm is used to convert attachments to a single form. One of the most widely known and easy to use algorithm developed for the English language is the Porter Stemmer root discovery algorithm implemented in this project [13]. Stemming algorithms used in NLP are given by Figure 3 [14].

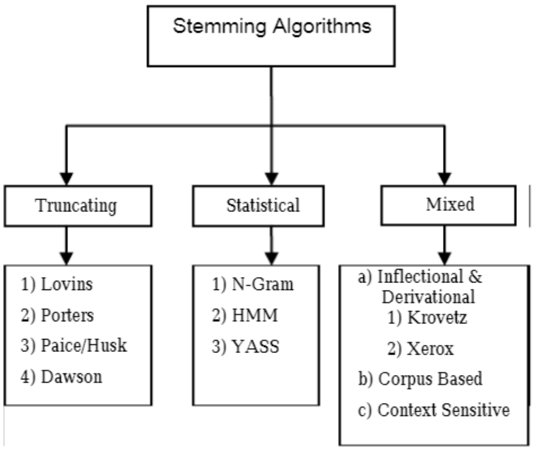


Fig. 3. Types of stemming algorithms

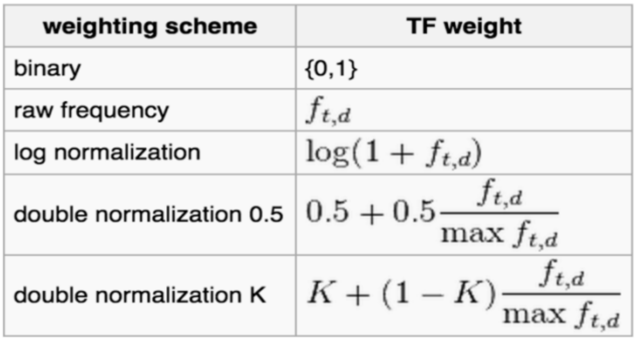
3.2 Weighting words

Keyword selection is applied in order to determine the meaningful words after pre-processing. There are different keyword selection methods. Helmholtz-based Gestalt Human perception theorem [15; 16], and TF-IDF [17], metrics are the most used.

In the study, the documents presented as seed or obtained as a result of web search are subjected to preprocessing step. Then, meaningful words are determined from these words. TF-IDF metric is used for the determination of meaningful words. The top n words with the highest TF-IDF meaning value are added to the dictionary and then searched on the web via these words added to the dictionary.

TF-IDF metric is the weighting factor calculated by statistical methodology, which shows the importance of a term in a document. It is used for statistical analysis on the texts which are the common application of natural language processing and text mining. It is also often referred to as a ranking algorithm under topics such as information retrieval.

Term Frequency (TF): A method used to calculate term weights in a document. The methods used to calculate the weight with TF are described in Figure 4.



weighting scheme	TF weight
binary	{0,1}
raw frequency	$f_{t,d}$
log normalization	$\log(1+f_{t,d})$
double normalization 0.5	$0.5+0.5\frac{f_{t,d}}{\max f_{t,d}}$
double normalization K	$K+(1-K)\frac{f_{t,d}}{\max f_{t,d}}$

Fig. 4. Term frequency weight calculation methods

Inverse Document Frequency (IDF): Measures the rank of the specific word for its relevancy within the text. Stop words which contain unnecessary information such as "a", "into" and "and" carry less importance in spite of their occurrence.

To explain the TF-IDF metric through the example;

Let's have a 1000-word document named "D1". Assuming that the word "Computer" is mentioned 10 times in this document, the TF value of the word "Computer" is calculated as follows.

TF = 10/1000 = 0, 01

We have 20 documents. Assuming that 10 of these documents refer to the word "Computer", the IDF value of the word "Computer" is calculated as follows.

IDF = log (10/5) = 0, 69

The TF-IDF calculation is equal to the multiplication of the TF and IDF values.

TF-IDF = 0, 01 * 0, 69= 0, 0069

3.3 *Obtaining documents from web search*

Among the keywords obtained using TF-IDF metric, the top n words with the highest value are added to the dictionary. Then, web search is repeated with the new words added to the dictionary and a list of documents is resulted from the web search. For this purpose, 3 different search engine APIs, Google, Bing and Yandex are experimented and generally different list of documents are returned from these search engines API's. These documents are then given back to the system and dictionary words are obtained. Then, the similarity rates of the dictionaries obtained are evaluated comparatively.

As a result of the implementation of TF-IDF metric, let the words "sport", "play" and "game" have the highest value. These words are then given as parameters to the relevant search engine API. Assuming that Google search engine API is used, meaningful words are combined with the space character and given to the Google search engine API as a parameter. For example, the words "sport", "play" and "game" are searched on the web as "game football sport". This process continues until the number of dictionary words reaches the specified parameter value. Thus, a continuous cycle is provided in the system.

3.4 *Dictionary similarity detection*

In this step, it is tried to determine the similarity of the dictionary obtained as a result of the application with the document/s presented initially. SimHash (Similarity Summary) algorithm was used as the similarity evaluation method.

The SimHash algorithm comes from Moses Charikar's paper [18] and is the basis for feature extraction originally designed to solve the deduplication tasks of hundreds of millions of web pages. The main idea is to map higher dimensional feature vectors to lower dimensional feature vectors. The implementation process is basically divided into five steps: namely segmenting, hash, weighting, merging, and descending dimension [19]. The working procedure that applies SimHash to generate a document to a 64-bit fingerprint is illustrated in Figure 5 [20].

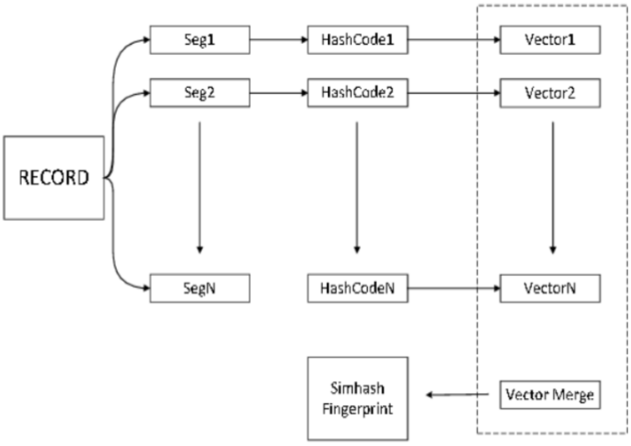


Fig. 5. The process of the simhash

SimHash algorithm has been used in many different studies for similarity detection. In Jiang [21], and Pi [22], studies, SimHash algorithm was used to obtain document similarity.

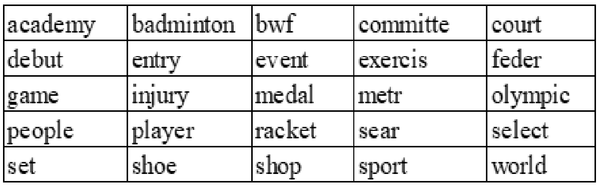
4 Experiments

For all dictionaries, documents belong to the sports data used in the work [23], is given to the system as a starting point. In the study, 3 different experiments were performed. The first and second experiments reveal the effect of the number of meaningful words added to the dictionary in each cycle, while the third experiment points out the effect of the change in the dictionary size on the dictionary performance rate.

4.1 *Experiment I*

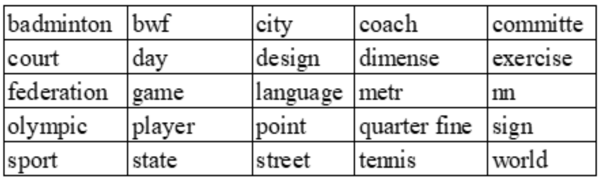
In this section, a sports (badminton) document is given to the system initially and 25 words dictionary is created. First of all, the dictionaries for 3 search engines: Google, Yandex and Bing were created differently. Later, hash similarities of the dictionaries were compared. Only the best document was selected from the result document list of web search engine and it was used as next iteration seed document. At the same time, only one word was added to the dictionary. This word, which was added to the dictionary, was then given as a parameter to the relevant API in the next web search.

Figure 6, 7, 8 gives the words of the dictionaries obtained by using Google, Bing and Yandex search engine APIs respectively.



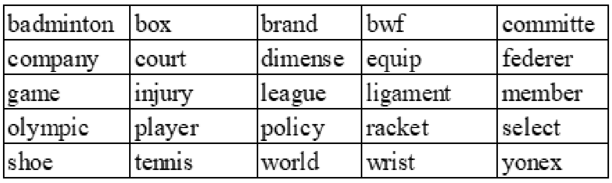
academy	badminton	bwf	committe	court
debut	entry	event	exercis	feder
game	injury	medal	metr	olympic
people	player	racket	sear	select
set	shoe	shop	sport	world

Fig. 6. 25 words limited dictionary initialized by only 1 sports document using google search api



badminton	bwf	city	coach	committe
court	day	design	dimense	exercise
federation	game	language	metr	nn
olympic	player	point	quarter fine	sign
sport	state	street	tennis	world

Fig. 7. 25 words limited dictionary initialized by only 1 sports document using bing search api



badminton	box	brand	bwf	committe
company	court	dimense	equip	federer
game	injury	league	ligament	member
olympic	player	policy	racket	select
shoe	tennis	world	wrist	yonex

Fig. 8. 25 words limited dictionary initialized by only 1 sports document using yandex search api

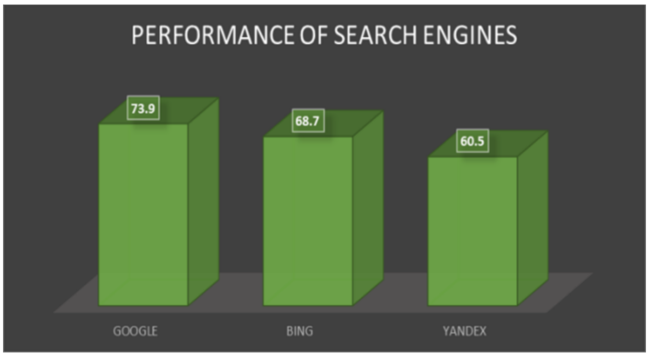


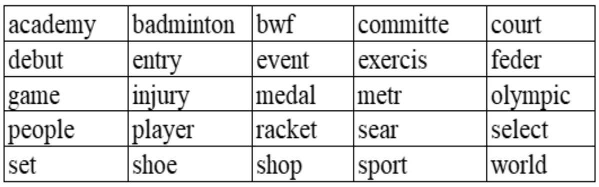
Fig. 9. Hash similarity performance of 25 words dictionaries based on search engines obtained by using only 1 sports document

Commenting on the Figure 9, it can be easily expressed that the hash similarity performance (in other words the quality of dictionary) of the dictionary obtained using the Google search engine API overwhelms the other dictionaries. The performance order of web search engines is related to the ranking algorithm used by the search engines. We can infer from the results in Figure 8. Google finds more relevant documents than Bing and Yandex, and performs better in terms of reaching purpose-oriented information.

4.2 Experiment II

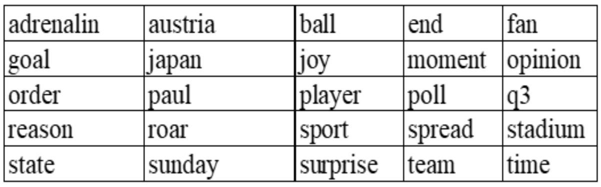
In this section, 2 sports (football) documents are given to the system initially and 25 words dictionary is created. Similar to Experiment I, dictionaries for 3 search engines were created differently. Later, hash similarities of the dictionaries were compared. For new iteration, two documents were obtained from the result list of web search engine and these were used as next iteration seed document. At the same time, only one word was added to the dictionary. This word, which was added to the dictionary, was then given as a parameter to the relevant API in the next web search.

Figure 10, 11, 12 gives the words of the dictionaries obtained by using Google, Bing and Yandex search engine APIs respectively.



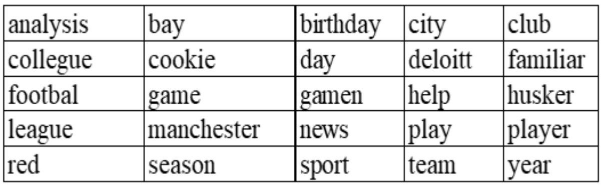
academy	badminton	bwf	committe	court
debut	entry	event	exercis	feder
game	injury	medal	metr	olympic
people	player	racket	sear	select
set	shoe	shop	sport	world

Fig. 10. 25 words limited dictionary initialized by 2 sports documents using google search api



adrenalin	austria	ball	end	fan
goal	japan	joy	moment	opinion
order	paul	player	poll	q3
reason	roar	sport	spread	stadium
state	sunday	surprise	team	time

Fig. 11. 25 words limited dictionary initialized by 2 sports documents using bing search api



analysis	bay	birthday	city	club
collegue	cookie	day	deloitt	familiar
footbal	game	gamen	help	husker
league	manchester	news	play	player
red	season	sport	team	year

Fig. 12. 25 words limited dictionary initialized by 2 sports documents using yandex search api

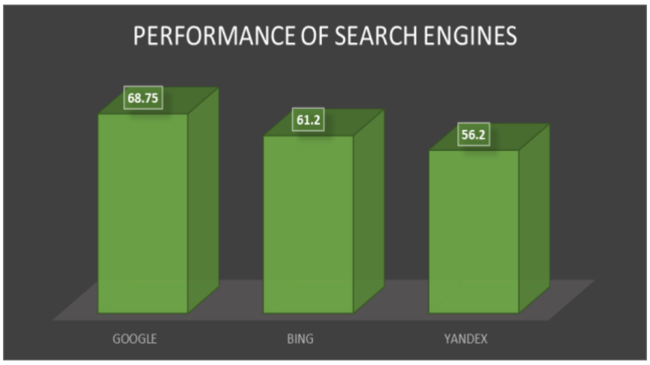


Fig. 13. Hash similarity performance of 25 words dictionaries based on search engines obtained by using 2 sports documents

Figure 13 shows the hash similarity performance of the dictionaries on the basis of search engines. When the results were examined, parallel to the results obtained in Experiment I was obtained. In both experiments, the performance of the dictionary obtained by using Google search engine is higher than other dictionaries. However, it is noted that the Google search engine performance decreases from 73.9% to 68.75%. Decrease in performance in the dictionaries created with other search engines can easily be seen in the similar way. In general, we can conclude that when the number of initial documents is increased while the number of dictionary words remaining constant, the hash similarity performances of dictionaries decrease. In other words, if the number of initial documents increases then the subject of the dictionary cannot be determined exactly. For this reason, it lets to the addition of unrelated words to the dictionary and so that causes deviations from the dictionary content.

4.3 *Experiment III*

In this section, a sports (badminton) document is given to the system initially and 50 words dictionary is created. Similar to Experiment I, dictionaries for 3 search engines were created differently. Later, hash similarities of the dictionaries were compared. Only the best document was obtained from the result list of web search engine and this was used as next iteration seed document. At the same time, only one word was added to the dictionary. This word, which was added to the dictionary, was then given as a parameter to the relevant API in the next web search. Unlike Experiment I, only the number of dictionary words was increased. When the number of dictionary words is increased, it will be observed how the performance rates will change.

Figure 14, 15, 16 gives the words of the dictionaries obtained by using Google, Bing and Yandex search engine APIs respectively.

academy	arcadia	area	badminton	blais
box	brand	bwf	committe	company
cookie	country	court	cutter	dark
debut	dimense	enjoy	entry	equip
event	excel	federer	game	injury
league	ligament	medal	member	metr
olympic	player	policy	postage	racket
school	sear	select	service	set
ship	shoe	shop	sport	technique
tennis	wayn	world	wrist	yonex

Fig. 14. 50 words limited dictionary initialized by only 1 sports document using google search api

academy	article	assist	associate	badminton
book	box	bwf	champion	captain
championship	committe	competition	cookie	court
cup	cutter	description	draw	editor
entry	event	federer	feedback	game
group	klcc	list	medal	member
metr	nn	olympic	play	player
point	policy	racket	read	select
service	set	shoe	side	singapore
sport	team	tennis	tournament	world

Fig. 15. 50 words limited dictionary initialized by only 1 sports document using bing search api

academy	associate	badminton	build	bwf
card	club	committe	cookie	country
court	cutter	debut	double	draw
entry	equip	event	excel	fat
feder	format	game	guest	hotel
image	injury	league	ligament	medal
metr	olympic	opponent	partner	player
policy	procedure	read	sear	service
set	shop	singapore	sport	team
technique	weight	world	wrist	year

Fig. 16. 50 words limited dictionary initialized by only 1 sports document using yandex search api

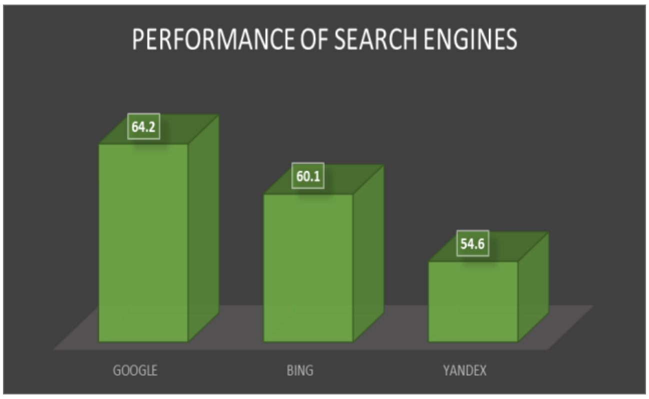


Fig. 17. Hash similarity performance of 50 words dictionaries based on search engines obtained by using only 1 sports document

Figure 17 shows the hash similarity performance of the dictionaries on the basis of search engines. When the results were examined, it is seen that the performance rank of search engines has not changed. However, when compared proportionally, the hash similarity performances of dictionaries in that experiment are less according to the results obtained in experiment 1 and experiment 2. In this experiment, only the effect of increasing the number of words in the final dictionary is evaluated. So that, the number of initial documents and their subject left constant, while the number of dictionary words were increased to 50. When the results were examined, although the Google search engine hash similarity performance is still better, performance is decreased dramatically from 73.9% to 64.2%. In general, we can conclude that when the number of dictionary words is increased, the dictionary quality decreases. In order to maintain the quality of bigger dictionaries, they should be checked periodically and the outlier words must be cleared.

5 Results

In the study, subject-specific dictionaries were created by processing the first document which suggested by web search engine. Three different search engine APIs, Google, Bing and Yandex, were used in order to determine the retrieval performance with respect to the relevance to the search. When Hash similarity of dictionaries has been compared, then the following inferences can be concluded.

- The highest performance in all experiments was achieved by using the Google search engine. Then, Bing and Yandex respectively.
- When the number of initial documents is increased, the dictionary similarity rate decreases a result of differences between given initial documents as expected, provided that the number and subject of the dictionary remain constant.
- When the number of dictionary words is increased, the dictionary quality decreases due to deviations occur in the dictionary.
- When the number of dictionary words is increased, the dictionary should be checked periodically. If there is deviation in dictionary, then corrections should be made manually (semi-automatic dictionary).
- The content of the document presented to the system is important. It should be informative enough about the topic without remaining any suspicion. Otherwise, dictionary words will not represent the subject of the desired dictionary.

6 Future Studies

As far as this model is considered, the following studies may be executed in the future.

- For meaningful word detection, the Helmholtz Principle or Rake algorithm can be used instead of TF-IDF metric.
- Other text similarity metrics (Cosine, Jaccard, etc.) could be used to measure performance of dictionaries.

References

[1] B V.Z. Kepuska and P. Rojanasthie, “Speech corpus generation from DVDs of movies and tv series,” Journal of International Technology and Information Management, vol. 20(1), pp. 49-82, 2011.

[2] R. Ellen, “Automatically constructing a dictionary for information extraction tasks,” Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 811-816, 1993.

[3] S. Koeva, I. Stoyanova, M. Todorova and S. Leseva, “Semi-automatic compilation of the dictionary of Bulgarian multiword expressions,” Proceedings of GLOBALEX 2016, pp. 86-95, 2016. https://doi.org/10.5281/zenodo.1469527

[4] K.E. Silverman, V. Anderson, J.R. Bellegarda, K.A. Lenzo and D. Naik, “Design and collection of corpus of polyphones and prosodic contexts for speech synthesis research and development,” Sixth European Conference on Speech Communication and Technology, PP. 5-9, 1999.

[5] A. Toprak, “Creating English dictionary with natural language processing,” Published Master Thesis, Istanbul Commerce University Institute of Science, Istanbul, 2019.

[6] C. Caldera, R. Berndt, E. Eggeling, M. Schröttner and D.W. Fellner, “PRIMA-towards an automatic review / paper matching score calculation,” The Sixth International Conference on Creative Content Technologies (CONTENT 2014), pp. 70-75, 2014.

[7] A. Mishra, and S. Vishwakarma, “Analysis of TF-IDF model and its variant for document retrieval,” International Conference on Computational Intelligence and Communication Networks (CICN), pp. 772-776, 2015. https://www.doi.org/10.1109/CICN.2015.157

[8] J. Lavid, H.J. Arús, B. Clerck and V. Hoste, "Creation of a high-quality, register-diversified parallel (English-Spanish) corpus for linguistic and computational investigations," 7th International Conference on Corpus Linguistics (CILC2015), vol. 198, pp. 249-256, 2015. https://doi.org/10.1016/j.sbspro.2015.07.443

[9] S.H. Sarkar and K. Mumit, "Automatic bangla corpus creation," PAN Localization Working Papers, vol. 3(1), pp. 22-26, 2010.

[10] B. Megyesi, J. Nasman and A. Palmer, "The uppsala corpus of student writings: corpus creation, annotation, and analysis," Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 3192-3199, 2016.

[11] F. Famili, W. Shen, R. Weber and E. Simoudis, "Data preprocessing and intelligent data analysis," Intell. Data Anal, vol. 1(4), pp. 3-23, 1997. https://doi.org/10.1016/S1088-467X(98)00007-9

[12] V. Agarwal, "Research on data preprocessing and categorization technique for smartphone review analysis," International Journal of Computer Applications, vol. 131(4), pp. 30-36, 2015. https://www.doi.org/10.5120/ijca2015907309

[13] C. Moral, A. Antonio, R. Imbert and J. Ramirez, "A survey of stemming algorithms in information retrieval," Information Research: An International Electronic Journal, vol. 19(1), pp. 76-80, 2014.

[14] R. Khoury, L. Shi and A. Hamou-Lhadj, "Key elements extraction and traces comprehension using Gestalt Theory and the Helmholtz Principle," 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 478-482, 2016. https://www.doi.org/10.1109/ICSME.2016.24

[15] B. Dadachev, A. Balinsky, H. Balinsky and S. Simske, "On the Helmholtz Principle for data mining," Third International Conference on Emerging Security Technologies, pp. 99-102, 2012. https://www.doi.org/10.1109/EST.2012.11

[16] S. Jabri, A. Dahbi, T. Gadi and A. Bassir, "Ranking of text documents using TF-IDF weighting and association rules mining," 2018 4th International Conference on Optimization and Applications (ICOA), pp. 1-6, 2018. https://www.doi.org/10.1109/ICOA.2018.837057

[17] A.G. Jivani, "A comparative study of stemming algorithms," Int. J. Comp. Tech. Appl, vol. 2(6), pp. 1930-1938, 2011.

[18] M.S Charikar, "Similarity estimation techniques from rounding algorithms," In Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, pp.380-388,2002. https://www.doi.org/10.1145/509907.509965

[19] Y. Li, F. Liu, Z. Du and D. Zhang, "A simhash-based integrative features extraction algorithm for malware detection," Algorithms-Open Access Journal, vol. 11(8), pp. 1-13, 2018. https://doi.org/10.3390/a11080124

[20] Y. Zhang, Z. Jin, W. Mu and W. Wang, "Research of distinct algorithm of short text based on simhash," DEStech Transactions on Engineering and Technology Research, pp. 120-126, 2017. https://www.doi.org/10.12783/dtetr/oect2017/16127

[21] Q. Jiang and M. Sun, "Semi-supervised simhash for efficient document similarity search," The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, vol. 1, pp. 93-101, 2011.

[22] B. Pi, S. Fu, W. Wang and S. Han, "SimHash-based effective and efficient detecting of near duplicate short messages," Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT '09), pp. 20-25, 2009.

[23] M. Turan and S. Ogtelik, "İngilizce dokümanlarda tema ve alt kavramlar tespit modeli," Düzce Üniversitesi Bilim ve Teknoloji Dergisi, vol. 6(4), pp. 754-764, 2018. https://doi.org/10.29130/dubited.420104