Generalized Cost Effective Automatic Dictionary Creation Using the Big Data on the Web

***Abstract—*** It is very important to obtain meaningful information from big data nowadays. At this point, the domain specific dictionaries help us. By the way, the researches in the language lexicography have been focused on automatic dictionary creation lately. In this work, a new approach is proposed to create a domain specific dictionary without any interference. Moreover, the dictionary can be extended later without any initialization cost.

First of all, an English document(s) related to the specific domain is given in order to initialize the process, called reference document(s). The meaningful words representing the reference document(s) are identified using TF-IDF values. The first dictionary words are obtained from the meaningful words of the reference document(s), call seed words. Then, Web search is applied by the Azure Web Cognitive Web Search system repetitively, constructing a query with the meaningful words obtained from the last iteration. The meaningful words found at each iteration are not added directly to the dictionary. Instead of this, WordNet dictionary is used to evaluate the similarity of each meaningful word within the dictionary, so that adding meaningless words to the dictionary is prevented. The meaningful words with higher similarity value above a certain threshold value (experimentally learned) are added to the dictionary until the desired number of words for the dictionary is reached. Although, this technique is applied to English words, it can be easily applicable to the all languages dictionaries. In order to measure the performance of the dictionary, the Hash similarity value was calculated. Dictionary up to 75, 2% hash similarity was generated by tests executed on reference documents with different subjects.

***Keywords—*** Automatic Dictionary Creation, Big Data, Hash Similarity, WordNet

**1. INTRODUCTION**

It is very important to obtain meaningful information from big data nowadays. At this point, the domain specific dictionaries help us. By the way, ​​the researches in the language lexicography have been focused on automatic dictionary creation lately. These dictionaries can be used for search engine optimization [1], automatic summarizing systems [2,3], theme determination [4] and text classification [5] research. For this purpose, it was aimed to determine the meaningful words of these documents by processing the documents and eliminating the words that are not important for us, so that the dictionary is created for the desired purpose. Thus, it is ensured that information can be represented more accurately.

The dictionary creation process can be done manually [6], semi-automatic [7] and automatic [8]. By hand-created dictionaries are become static and they require continuous external intervention to grow the dictionary, results in the maintenance cost. It is important to automate this process in order to get rid of this maintenance cost and to obtain a more generic application.

Ellen R. [8] mentioned the shortcomings of manual dictionaries. Knowledge-based natural language processing systems are often criticized, though they have achieved good success with specific tasks. Because of they are dependent on a specific-domain dictionary that requires manual knowledge engineering. This knowledge engineering bottleneck is not practical for implementing knowledge-based NLP systems in real-world applications. Thus, they cannot be easily scaled or transferred to the new fields.

In this study, an algorithm is proposed and experimentally tested to create an automatic dictionary for a specific domain. The initial step is to find meaningful words from the document(s) given by the user, called reference document(s). The meaningful words representing the reference document(s) are identified using TF-IDF values. The first dictionary words are obtained from the meaningful words of the reference document(s), call seed words. Then, Web search is applied by the Azure Web Cognitive Web Search system repetitively, constructing a query with the meaningful words obtained from the last iteration. The meaningful words found at each iteration are not added directly to the dictionary. Instead of this, WordNet dictionary is used to evaluate the similarity of each meaningful word within the dictionary, so that adding meaningless words to the dictionary is prevented. The meaningful words with higher similarity value above a certain threshold value (experimentally learned) are added to the dictionary until the desired number of words in the dictionary is completed. Thus, the dictionary will continue growing automatically and continuously without any intervention from the outside.

Hash Similarity method was used to measure the overall similarity success of the project. The notion of locality-sensitive hashing was ﬁrst introduced by Indyk et al. [9]. For efﬁcient near neighbor search, the locality-sensitive hashing (LSH) techniques exploit special hash functions which make buckets contain similar keys (data), yet do not guarantee that all data in a bucket are similar each other. The so-called locality-sensitive hash functions provide high probability for similar data to be in the same bucket, but low probability for dissimilar data to be in the same bucket [10].

In the second section of the article, prior works in the field of corpus generation were discussed. The methods and studies used for dictionary creation are explained in the third section. The data sets mentioned in the third section are explained in the fourth section. Finally, results are discussed and evaluated in the last section.

**2. PRIOR WORK**

From the past to the present, studies have been carried out about the creation of the dictionary with different methods and analyzes. These studies [9,11] especially were started between 1990 and 2000 and it continues up to date with different techniques in line with the changing needs. The purpose of most of these studies is to obtain the desired information from the bulk data.

In our study, important words belonging to documents were found and the documents were tried to be represented with these words. In this study, TF-IDF metrics which is the most commonly used method for determining important words of the documents are used. TF-IDF metrics have been successfully applied in many different natural language processing applications. One of these important studies is Jiaul H. Paik's [12] effective ranking study in 2013 using TF-IDF metrics. In the study, Paik addressed the limitations of the pivoted length normalisation by exploiting new statistical factors in the Multi Aspect TF (MATF) schema. One component of the term frequency is effective for short queries, while the other performs better on long queries. The final weight is then measured by taking a weighted combination of these components, which is determined on the basis of the length of the corresponding query. Experiments conducted on a large number of TREC news and web collections demonstrate that the proposed scheme almost always outperforms five state of the art retrieval models with remarkable significance and consistency.

In another TF-IDF term weighting study [13], an SMS model was developed in order to determine the statistical importance of a word for an SMS categorization. First, all SMSs are converted to text documents. After the pre-processing of the SMSs, the vector space model is prepared and a weight is given to each term. This weighting method predicts the significance of a word for an SMS classification problem. Experiments reported in the article show that this weighting method significantly improves the classification accuracy measured in many categorization tasks.

TF-IDF metrics are also used in sentiment analysis studies [14]. Das and his team propose a technique for text sentiment classification using term frequency-inverse document frequency (TF-IDF) along with Next Word Negation (NWN). They have also compared the performances of binary bag of words model, TF-IDF model and TF-IDF with “next word negation” (TF-IDF-NWN) model for text classification. The achieved results show significant increase in accuracy compared to earlier methods.

Qaiser et al. [15] used TF-IDF metrics in their study in 2018. In the study, the document is determined whether the word is related to the document. In this model, TF-IDF metric is used to determine the weights of words and grouping is done by this weight value. According to the results, the relationship of a word with the document has been determined.

As stated in the previous section, the dictionary creation process is operated in three different ways; automatic, semi-automatic and manual. In 1993, Riloff Ellen conducted a study [8] on automatic dictionary creation. In his work, Riloff Ellen developed a system called AutoSlog, a dictionary of domain specific concepts for extracting information from text. When AutoSlog is given a text, AutoSlog creates a series of dictionary entries by subtracting the requested information from this text. If the text provided to AutoSlog represents the requested information, the dictionary created by AutoSlog will yield significant success. AutoSlog dictionary with 5 person-hour, a dictionary containing the terrorist event was created. The AutoSlog dictionary was then compared to a handmade dictionary that was made by two talented graduate students, requiring approximately 1500 person-hour effort. As a result, the AutoSlog dictionary provided 98 % of the performance of the handmade dictionary.

In 1999, Silverman et al. [11] describe the design and structure of the Victorian dictionary created to support speech synthesis research and development at Apple Computer. The Victorian dictionary consists of 5 main chapters. These are polyphony, prosodic context, repetitive speech, function word sequences and continuous speech. This dictionary is designed to cover a particular aspect of each of the speech synthesis. The Victorian dictionary is generally created in US English. The purpose of the Victorian dictionary is the collection of semantic texts through speech. The dictionary is used in statistical estimation of the time and step models for MacinTalk 4, the next generation text-to-speech system from Apple.

Another automatic dictionary creation study [16] is the work of S. Vorapatratorn, A. Suchato and P. Punyabukkana. This study describes the method of automatic dictionary creation using a specific phonetic distribution. Usually, the system selects its data by downloading continuous text from the Internet through a web browser. The covetous algorithm is applied to text to extract the appropriate words, this process continues until the appropriate text dictionary is installed. The results of the study show that the number of data withdrawn from the internet can realize the target phonetic distribution and create a telephone coverage area of ​​99.13 %. This text dictionary can then be used to efficiently produce the speech dictionary.

Natalia Grabar, Vincent Claveau and Clement Dalloux [17] created semi-automatic a new corpus in French which provides medical data close to those produced in the clinical context: description of clinical cases and their discussion. Overall, the corpus currently contains over 397,000 word occurrences excluding punctuation marks. The corpus is currently annotated with several layers of information; linguistic (PoS-tagging, lemmas) and semantic (the UMLS concepts, uncertainty, negation and their scopes). The corpus will be enriched with more clinical cases published.

One example of creating an automatic dictionary is MirasText [18]. MirasText which is an automatically generated text corpus for Persian language is presented. In this study, over 250 Persian websites were crawled and several fields like content, description, keywords, title, etc have been extracted to generate MirasText. The generated corpus contains more than 2.8 million documents and more than 1.4 billion content words. MirasText is the largest Persian text corpus available which can be used for a variety of NLP applications like language modeling, automatic summarization, keyword extraction, title generation, etc.

In the study [6] of Kepuska Veton Z. and Rojanasthien, a data collection system is created to produce speech dictionary from movies, TV series and DVDs. The dictionary provides a lower cost solution compared with the traditional speech dictionary acquisition method. In addition, it was stated that the collection of data and processing the dictionary was shorter.

Studies on creating a language dictionary using natural language processing techniques can be applied to different languages. Although studies referring to the English language [8,11] are more numerous, there are also dictionary works in languages such as Tigrinya [19], Thai [20], French [17], Arabic [21, 22] and Turkish [23]. One of them is the WordNet study in Turkish. In this study [23], especially WordNet Turkish, a detailed WordNet literature research was conducted. In the long term, the widest Turkish information dictionary has been prepared with the related dictionaries prepared so far. Thus, it is stated that the dictionary of relational information terms, which is the basis of large informatics projects, is the largest dictionary according to the other dictionaries and the multiplicity of the number of words.

In 2018, Vijay and his team [24] created a Hindi-English code mixed dictionary using tweets published online for the last 8 years. To create this dictionary, tweets were taken from Twitter, using the Twitter Python API, which primarily uses Twitter's advanced search option. All information such as received tweets, timestamp, URL, text, user, retweets, answers, full name, identity and likes were converted to json format. A comprehensive semi-automated procedure was performed to remove all noisy tweets. After all of these steps, a dictionary of 2866 words was created and dictionary words were classified as happiness, sadness, anger, surprise, hate and multiple emotion. Emotional analysis was performed for the tweets published online, and a 58.2 % accuracy was obtained.

**3. METHOD**

In order to identify the meaningful words of the documents fed into the system, the documents initially are pre-processed. After meaningful terms are determined, they are used to feed the dictionary and the following web search. The general system schema is given by Figure 1.

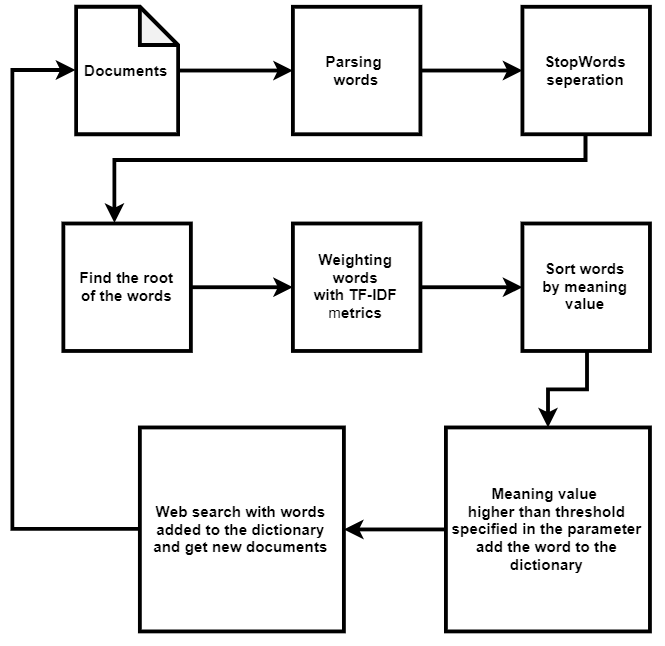


Figure 1. General structure of automatic dictionary creation

*3.1.* **Pre-processing**

Text pre-processing is one of the most important preliminary steps of natural language processing, often troublesome, but affecting the success of the algorithm. It is necessary to remove prepositions, conjunctions, exclamations, letters and words with no categorical meaning. These words, called stop words, must be filtered in the first step of text pre-processing [25].

Text pre-processing is applied to the each document using noise removal, lexicon normalization and object standardization techniques.

* Operations are performed on the noise reduction connectors.
* Dictionary includes operations on the normalized formation of words from the same root.
* Object standard formation is pre-processing techniques which can be done on abbreviations [26].

After this process, the words of the documents are converted to the lower case format. Then, it is necessary to separate the words from the suffixes and find their root form. For this purpose, stemming is applied to words. The main purpose of stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form. The most common rule-based stemming algorithm for the English language was developed by Porter and it is still most commonly used method.

The most commonly known and easy-to-use Porter Stemmer algorithm [27] has been implemented in this project for the reason it has been successfully used in the previous studies [28] of stemming. Figure 2 lists the stemming algorithms in NLP.

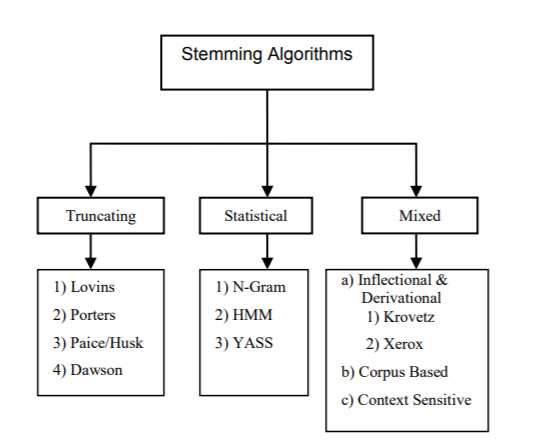


Figure 2. Types of stemming algorithms

*3.2.* **Selection of Keywords**

At this stage, the system was tried to find meaningful words for both the initial set of documents and the set of documents obtained as a result of the Web search. In this study, TF-IDF metric was applied to find the meaning values of words. TF-IDF consists of combination of two different concepts; Term Frequency and Inverse Document Frequency. The term frequency-inverse document frequency (TF-IDF) metric is commonly used to weigh each word in the text document according to how unique it is. In other words, the TF-IDF approach captures the relationship between words, text documents and specific categories [29].

Let’s suppose, we have a document “T1” containing 5000 words and the word “Alpha” is present in the document exactly 10 times. It is very well-known fact that, the total length of documents can vary from very small to large, so it is highly possible that a term may occur more frequently in large documents in comparison to small documents. So, in order to normalize document length, the occurrence of any term in a document is divided by the total terms present in that document, to find the term frequency. So, in this case the term frequency of the word “Alpha” in the document “T1” will be;

TF = 10/5000 = 0.002

Inverse Document Frequency (IDF) assigns less weight to common words and more weight to rare words. For example, if we have 10 documents and the term “technology”, is available in 5 of these documents, the IDF value is calculated as follows in.

IDF = log\_e (10/5) = 0.3010

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

TF-IDF = 0.002 \* 0.3010 = 0.000602

In the study, after the documents fed to the system are subjected to pre-processing steps, the number of words in the document is recorded in the database. Then, using the formulas mentioned above, TF, IDF and TF-IDF values of each word are calculated. The words obtained by applying TF-IDF metrics are recorded in the table given in Figure-3. After calculating these values, the words with a meaning value above the value determined as the parameter are added to the dictionary. The dictionary hash similarity values obtained as a result of adding different threshold meaning values are given by detail in section-4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DocumentId | Word | TF | IDF | TF\_IDF |
| 1 | season | 0.0422 | 1 | 0.0422 |
| 1 | yard | 0.0578 | 0.6931 | 0.0400 |
| 1 | pass | 0.0533 | 0.6931 | 0.0370 |
| 1 | player | 0.0222 | 1 | 0.0222 |
| 1 | game | 0.0200 | 0.6931 | 0.0139 |
| 1 | touchdown | 0.0200 | 0.6931 | 0.0139 |
| 1 | football | 0.0133 | 1 | 0.0133 |

Figure 3. Table where TF-IDF metrics are applied

*3.3* **Creating the First Dictionary Words from the Starting Document**

When the document/s related to the subject of the desired dictionary is given to the system for the first time, the meaningful words of these initial seed document/s are determined using TF-IDF metrics. Then, as can be seen in the table structure mentioned in Figure-3 the highest TF-IDF value is added to the dictionary initially. Web search is then started by this Word. However, the rule of determining candidate meaningful words for insertion into the dictionary through Web search is not only the most meaningful word, but also the word/s which TF-IDF value/s are above 0,03.

*3.4.* **Determination of Similarity Value**

When candidate meaningful word/s selected after web search, then a similarity value is calculated for each word with respect to the dictionary using WordNet similarity. Finally, the words with similarity above the similarity threshold value are added to the dictionary. For example, after finding meaningful word for Web search results, suppose that 3 words with the highest TF-IDF value will be added to the dictionary and these 3 words are "student", "math" and "lesson". Let the words in the dictionary be “team“ and “ball”.

A loop is then used to calculate the WordNet similarity of each meaningful word with the words in the dictionary.

If we take the word “football;

WordNet Similarity (“football”, “team”) = 0,35

WordNet Similarity (“football”, “ball”) = 0,30

Total WordNet Similarity (“football”) = 0,35 + 0,30 =0,65 is calculated.

Then the average WordNet similarity value of the word is calculated. The following formula is used for this calculation.

Average WordNet Similarity = Total WordNet Similarity / Dictionary Word Count

According to this formula,

Average WordNet Similarity = 0,65 / 2 =0,325 is calculated.

It then decides whether the word should be added to the dictionary. If the average WordNet Similarity parameter value in the application is 0.35, the word is not added to the dictionary, and 0.30 is added to the dictionary.

WordNet is the English dictionary database in the Cognitive Science Laboratory of Pricenton University. Names, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each of which expresses a separate concept. antonymy, hypernymy, such as hypernymy, meaning a more abstract and general meaning of a word in the database, hypernymy feature is used in this database. For example, by looking for the word hypernym in the WordNet database of the word cat, the concept of this word can be determined as an animal [30]. WordNet is free and can be downloaded publicly. The structure of WordNet is therefore a useful tool for computational linguistics and natural language processing [31]. Figure 4 below shows the WordNet hierarchy.

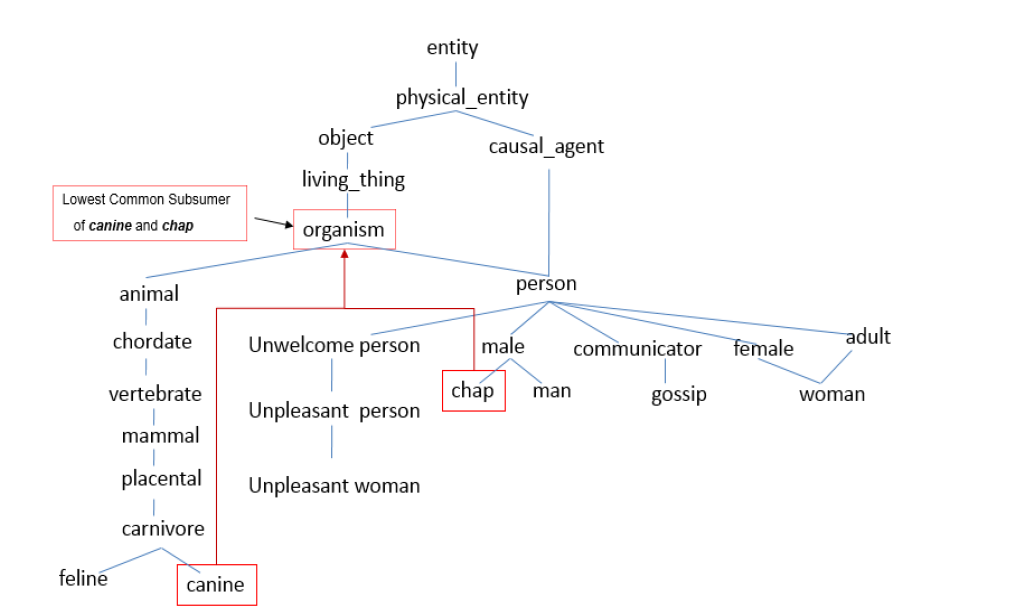


Figure 4. A sample WordNet hierarchy. [32]

*3.5.* **Web Search Process**

The most meaningful word inserted into dictionary for the last iteration is used in the next web search together with the synonym word obtained from WordNet. For example, if the word added to the dictionary is "student", the synonym for "student" is "pupil". In the web search, "student" and "pupil" are used together.

The Azure Cognitive Web Search method is used to search on the Web. The words to be used in the web search are combined with the space character and given as a parameter to the Azure Web Search service. For example, the words are “ronaldinho”, “barcelona” and “el classico” respectively, and these words are used together as ”ronaldinho barcelona el classico” on the web search. Web search with these words will result in more than one document. Among these documents, a single document that has not been previously processed is selected. Then, the pre-processing steps are applied to this document and new meaningful words are obtained. Thus, the dictionary was constantly expanded with new documents and the dictionary was prevented repeating itself. This process continues until the dictionary reaches the number of dictionary words which a parameter value is given initially by the user to the system. Thus, a continuous cycle is provided in the system.

*3.6***. Determination of General Similarity Value of Dictionary**

In this step, the similarity of the dictionary is determined by SimHash algorithm. The SimHash algorithm is an algorithm used to find similarity between files or websites in applications such as search engine, especially in text processing. The SimHash algorithm sees two files as vectors and tries to find the cosine link between these vectors (vector, vector).

SimHash is a hash function, and the more similar its property is to the text input, the smaller the Hamming distance of the hash values (Hamming distance - the number of locations where the corresponding symbols are different). The algorithm works by dividing the text into pieces and combining each piece with a function of your choice. Each mixed set is represented as a binary vector and the bit value is converted to +1 or -1 depending on whether the bit value is 1 or 0. To get SimHash, we collect all bit vectors as bit-wise. Finally, if the total is negative, the bits that result to 1 otherwise are set to 0.

SimHash algorithm has been used in many different studies for the purpose of similarity detection. For example, in the Jiang and Pi study [33, 34], SimHash algorithm is used to obtain document similarity. Figure 5 below illustrates the working procedure of the SimHash algorithm.

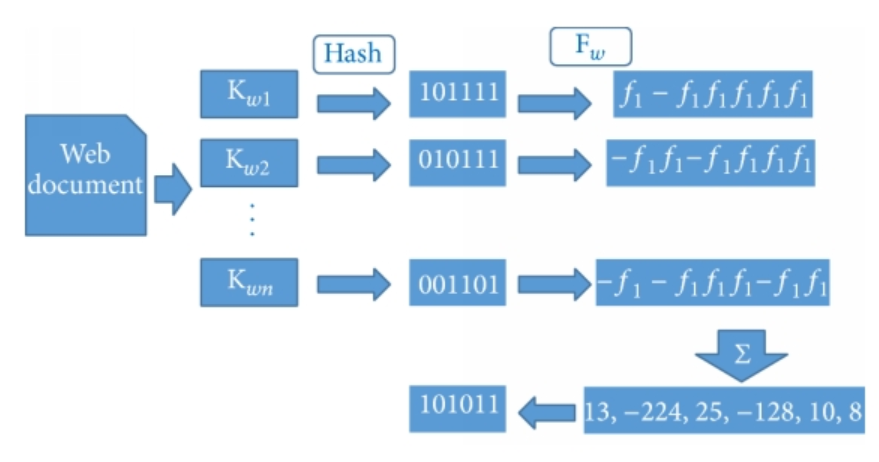


Figure 5. Working procedure of the SimHash algorithm [35].

**4. EXPERIMENTS**

For all dictionaries, documents belongs to the sports data used in the work [36] is given to the system as a starting point. Documents containing html content are fed into the system. First of all, the number of dictionary words is experimented by 25, 50 and 100 in general. Additionally, the effect of the meaning value of a word calculated by the TF-IDF metric is evaluated for candidate word selection for dictionary. For each experiment, the dictionary similarity values are calculated and discussed.

**4.1. Experiment I**

In Figure 6 below, the hash similarity value for 25-words limited dictionary initialized by only 1 sport (badminton) document is given.

|  |  |
| --- | --- |
| **Parameter Name** | **Parameter Value** |
| TF-IDF Meaning Value to Use to Add Word to Dictionary | 0,03 |
| Dictionary Maximum Number of Words | 25 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | **75,2** |

Figure 6. Hash similarity value for 25-words limited dictionary initialized by only 1 sports document

Similarity value of 25 words dictionary is higher than the dictionaries of 50 and 100 words. The most important reason for achieving this similarity value is that new documents obtained from Web for each iteration are deviating from the original reference document topic. Figure 7 lists the words obtained for parameters given in Figure 6.

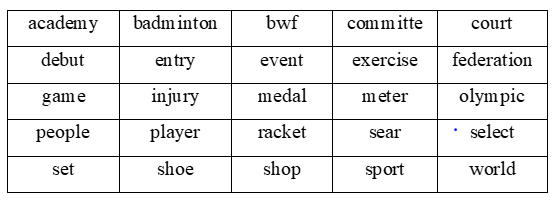


Figure 7. 25-words limited dictionary initialized by only 1 sports document

In Figure 8 below, the hash similarity value for 50-words limited dictionary initialized by only 1 sport (badminton) document is given.

|  |  |
| --- | --- |
| **Parameter Name** | **Parameter Value** |
| TF-IDF Meaning Value to Use to Add Word to Dictionary | 0,03 |
| Dictionary Maximum Number of Words | 50 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | **73,8** |

Figure 8. Hash similarity value for 50-words limited dictionary initialized by only 1 sports document

Similarity value of 50 words dictionary is higher than the dictionary of 100 words. Obviously, when all parameters are kept constant and only the number of dictionary words is increased, the dictionary similarity value decreases. Figure 9 lists the words obtained for parameters given in Figure 8.



Figure 9. 50-words limited dictionary initialized by only 1 sports document

In Figure 10 below, the hash similarity value for 100-words limited dictionary initialized by only 1 sport (badminton) document is given.

|  |  |
| --- | --- |
| **Parameter Name** | **Parameter Value** |
| TF-IDF Meaning Value to Use to Add Word to Dictionary | 0,03 |
| Dictionary Maximum Number of Words | 100 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | **68,7** |

Figure 10. Hash similarity value for 100-words limited dictionary initialized by only 1 sports document

Similarity value of 100 words dictionary is lower than the similarity values of dictionaries of 25 and 50 words. With this result, we can make an inference. When the maximum number of words in the dictionary is increased, deviations occur in the dictionary after a certain number of words. The factor that causes the deviation is that the Web search is done through this word after adding an incorrect word to the dictionary. In order to prevent deviations in the dictionary, the dictionary should be checked periodically and the words related to this document should be added to the dictionary based on the reference document given initially. Figure 11 lists the words obtained for parameters given in Figure 10.



Figure 11. 100-words limited dictionary initialized by only 1 sports document

**4.2. Experiment II**

By this experiment, the effect of the meaning value of a word is worked.. If the meaning value of a word calculated by the TF-IDF metric is higher than 0,03, then it is more likely a candidate for the dictionary. If this threshold is applied through selection of dictionary words, higher dictionary similarity values are obtained. On the other hand, in the reverse case, if the meaning value below 0,03 is selected for dictionary word selection, it results on the lower dictionary similarity values. On the following experiments, the threshold value for the TF-IDF meaning value is selected 0.2, and 25 and 50 words dictionaries are created respectively.

In the Figure 12 below, the hash similarity value for 25-words limited dictionary initialized by only 1 sport (badminton) document is given.

|  |  |
| --- | --- |
| **Parameter Name** | **Parameter Value** |
| TF-IDF Meaning Value to Use to Add Word to Dictionary | 0,02 |
| Dictionary Maximum Number of Words | 25 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | **62,5** |

Figure 12. Hash similarity value for 25-words limited dictionary initialized by only 1 sports document

One sports document (badminton) is again given to the system and as a result of the calculation of TF-IDF metrics, a 25-word dictionary is produced using the words with a meaning value equal or lower than 0,02. The overall similarity value of final dictionary is calculated by the Hash Similarity method and 62.5 % similarity is obtained. This similarity value is lower than the similarity value of all the dictionaries obtained by adding the words with the meaning value over 0,03. It is realized that addition of words lower than a meaningful threshold value to the dictionary causes deviations and decreases similarity of dictionary words. Figure 13 includes the words of the dictionary obtained by this experiment.

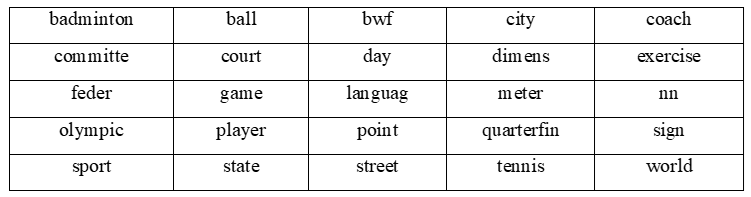


Figure 13. 25-words limited dictionary initialized by only 1 sports document

In Figure 14 below, the hash similarity value for 50-words limited dictionary initialized by only 1 sport (badminton) document is given.

|  |  |
| --- | --- |
| **Parameter Name** | **Parameter Value** |
| TF-IDF Meaning Value to Use to Add Word to Dictionary | 0,02 |
| Dictionary Maximum Number of Words | 50 |
| Number of Starting Documents | 1 |
| Dictionary Similarity Value % | **59,2** |

Figure 14. Hash similarity value for 50-words limited dictionary initialized by only 1 sports document

The overall similarity of this dictionary is calculated by Hash similarity method and it is equal to 59.2%. This similarity value is lower than the similarity of 25-word dictionary as expected a result of Experiment I. Figure 15 includes the words of the dictionary obtained by this experiment.



Figure 15. 50-words limited dictionary initialized by only 1 sports document

**5. RESULT**

In the work, meaningful words of the seed document/s are calculated by using TF-IDF metric and a specific domain dictionary is created automatically. Through this study, the effect of determining a threshold for the meaningful value of a word in order to add to the dictionary, and dictionary size are researched additionally. As a result, we reveal two realities. First one is that when the dictionary size grows, then similarity of dictionary decreases as expected intuitively. The explanation of this result can be given by that there is less deviation in the dictionary because the number of dictionary words is small. Second one is that bigger threshold for the meaningful value of a word in order to add to the dictionary enhances the dictionary similarity. This is also a result of selecting more meaningful (similar) words.

Compared to similar studies, it is seen that the results of the study are quite successful in terms of average results, dictionary creation and growth rate. Compared to the similar study [24] mentioned in the literature review section, it has achieved successful results in automatic dictionary creation in 2 studies. However, unlike [24], the Web search section is included in this study. This study is valuable than the study of [24] in terms of working on instant and current data.

WordNet similarity calculation is used in this study in order to determine whether the words to be added to the dictionary are related to the subject to be created. In general, the following conclusions can be drawn from the results of the study.

* The set of documents should be related to the dictionary you want to create. If the system is fed with a set of documents that are not associated with the desired dictionary, the resulting dictionary will contain meaningless words.
* Once the meaningful words are found, the words with a similarity value are added to the dictionary. When the threshold value is increased, the speed of dictionary creation decreases, while the similarity value of the final dictionary is increasing.
* Once the meaningful words are found, the ratio of these meaningful words as the only parameter is used in web search. Therefore, if the percentage value to be used in the Web search part of the meaningful words changes, the success rate will change.
* Querying with the last words added to the dictionary for processing with different data continuously during the web search process will prevent duplication of the dictionary. It is therefore useful to mark the last words in the dictionary.

**6. FUTURE WORKS**

The following studies, which are thought to contribute to the literature, will be discussed.

* Since the suggested system operates in a continuous growth, the number of iterations in which the system approaches saturation can be determined.
* WordNet similarity method was used to calculate the similarity ratio between words in the dictionary. According to the WordNet similarity result, it is decided whether to add the word to the dictionary. Different similarity calculation techniques can also be examined. For example, Word2vec similarity method and WordNet similarity method can be compared. Then, the structural differences of the two dictionary can be revealed.
* Once you have determined the word to add to the dictionary, all the synonyms words of that word can be added to the dictionary, together with this word. SimHash algorithm can be calculated by comparing the results.

**REFERENCES**

[1] S. Vorapatratorn, A. Suchato, P. Punyabukkana, "Automatic online text selection for constructing text corpus with custom phonetic distribution",Ninth International Conference on Computer Science and Software Engineering (JCSSE), Bangkok, 2012, pp. 6-11, 2012.

[2] Hovy, Eduard & Lin, Chin-Yew. (2001). Automated text summarization and the SUMMARIST system. Proceedings of a Workshop on Held at Baltimore. 10.3115/1119089.1119121.

[3] Gholamrezazadeh, Saeideh & Salehi, Mohsen & Gholamzadeh, Bahareh. (2010). A Comprehensive Survey on Text Summarization Systems. 1 - 6. 10.1109/CSA.2009.5404226.

[4] Habacha, Anja & Kboubi, Ferihane & Ben Ahmed, Mohamed. (2011). Thematic Analysis And Visualization Of Textual Corpus. International Journal of Computer Science and Engineering Survey. 2. 10.5121/ijcses.2011.2402.

[5] Nguyen, Thien & Shirai, Kiyoaki. (2013). Text Classification of Technical Papers Based on Text Segmentation. 10.1007/978-3-642-38824-8\_25.

[6] V. Z. Këpuska, P. Rojanasthien, "Speech Corpus Generation from DVDs of Movies and TV Series", Journal of International Technology and Information Management: Vol. 20 : Iss. 1 , Article 4, 2011.

[6] Al-Talib, Ghayda A., and Hind S. Hassan, "A Study on Analysis of SMS Classification Using TF-IDF Weighting", International Journal of Computer Networks and Communications Security (2013), Vol.1, No.5, pp.189 – 194.

[7] Koeva, Svetla & Stoyanova, Ivelina & Todorova, Maria & Leseva, Svetlozara. (2016). Semi-automatic Compilation of the Dictionary of Bulgarian Multiword Expressions. Proceedings of GLOBALEX 2016: Lexicographic Resources for Human Language Technology, Workshop at LREC2016, Portorož, Slovenia.

[8] R.Ellen,“Automatically Constructing a Dictionary for Information Extraction Tasks”, Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI Press / MIT Press, pages 811–816, 1993.

[9] P. Indyk and R. Motwani, “Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality,” Proc. of STOC, 1998

[10] Lee, Keon Myung. (2012). Locality-Sensitive Hashing Techniques for Nearest Neighbor Search.

International Journal of Fuzzy Logic and Intelligent Systems. 12. 300-307. 10.5391/IJFIS.2012.12.4.300.

[11] K.E. Silverman, V. Anderson, J.R Bellegarda, K.A. Lenzo ,D. Naik, “ Design and collection of a corpus of polyphones and prosodic contexts for speech synthesis research and development”, 1999.

[12] H. Paik, Jiaul. (2013). A novel TF-IDF weighting scheme for effective ranking. 10.1145/2484028.2484070.

[13] Al-Talib, Ghayda A., and Hind S. Hassan, "A Study on Analysis of SMS Classification Using TF-IDF Weighting", International Journal of Computer Networks and Communications Security (2013), Vol.1, No.5, pp.189 – 194.

[14] Das, Bijoyan & Chakraborty, Sarit. (2018). An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation.

[15] Qaiser, Shahzad & Ali, Ramsha. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications. 181. 10.5120/ijca2018917395.

[16] S. Vorapatratorn, A. Suchato, P. Punyabukkana, "Automatic online text selection for constructing text corpus with custom phonetic distribution",Ninth International Conference on Computer Science and Software Engineering (JCSSE), Bangkok, 2012, pp. 6-11, 2012.

[17] Natalia Grabar, Vincent Claveau, Clément Dalloux.

CAS: French Corpus with Clinical Cases.

LOUHI 2018 - The Ninth International Workshop on Health Text Mining and Information Analysis,

Oct 2018, Bruxelles, France. pp.1-7. ⟨hal-01937096⟩

[18] Sabeti, Behnam & Abedi, Hossein &

Janalizadeh Choobbasti, Ali & H E Mortazavi Najafabadi,

S & Vaheb, Amir. (2018). MirasText An Automatically Generated Text Corpus for Persian.

[19] Hafte Abera., Sebsibe H/Mariam.,

“Design of a Tigrinya Language Speech Corpus for Speech Recognition , ” Association for Computational Linguistics,ss. 78–82, Santa Fe, New Mexico, USA ,2018

[20] S. Vorapatratorn, A. Suchato and P. Punyabukkana,

"Automatic online text selection for constructing text corpus with custom phonetic distribution," 2012 Ninth International Conference on

Computer Science and Software Engineering (JCSSE), Bangkok, 2012, pp. 6-11.

[21] Alansary, Sameh & Nagi, Magdy. (2014).

The International Corpus of Arabic: Compilation, Analysis and Evaluation. 8-17. 10.3115/v1/W14-3602.

[22] Alansary, Sameh & Nagi, Magdy & Adly, Noha & Alexandrina, Bibliotheca. (2008). Building an International Corpus of Arabic (ICA): Progress of Compilation Stage.

[23] Y.Aktaş , E.Y İnce, A. Çakır, A. Kutlu, “Wordnet ve Bilgisayar Ağ Terimleri Sözlüğünün Oluşturulması” Akademik Bilişim 2016, Adnan Menderes Üniversitesi, 30 Ocak- 5 Şubat, Aydın, 2016.

[24] Vijay D., Bohra A., Singh V., Akhtar S.S., Shrivastava M., (2018). Corpus Creation and Emotion Prediction for Hindi-English Code-Mixed Social Media Text. NAACL-HLT.

[25]Internet:O.Durmaz,http://www.osmandurmaz.com/makale/metinonisleme/makale.aspx, 20.06.2019.

[26] Internet: F.Filiz,Medium ,<https://medium.com/@fahrettinf/5-1-do%C4%9Fal-dil-i%CC%87%C5%9Fleme-natural-language-process-a92079c3d271>,20.06.2019.

[27] Porter M.F, Snowball: A language for stemming algorithms. 2001.

[28] Jivani, Anjali. (2011). A Comparative Study of Stemming Algorithms. Int. J. Comp. Tech. Appl.. 2. 1930-1938.

[29] Deepshikha Patel, Monika Bhatnagar, Mobile SMS Classification, International Journal of Soft Computing and Engineering (IJSCE), Volume-I, Issue-I, March 2011.

[30] Aydın, C.R., Erkan, A., Güngör, T., Takçı, H., Sözlük Kullanarak Türkçe için Kavram Madenciliği Metotları Geliştirme, Akademik Bilişim, 2014, Mersin.

[31]Internet:T.TKüçük,Medium,<https://medium.com/5bayt/wordnet-nedir-6910c6f98837>, 21.04.2019.

[32] Wei, Tingting & Chang, Huiyou. (2015). Measuring Word Semantic Relatedness Using WordNet-Based Approach. Journal of Computers. 21. 252-259. 10.17706/jcp.10.4.252-259.

[33] Jiang, Q., & Sun, M. (2011). Semi-Supervised SimHash for Efficient Document Similarity Search. ACL.

[34] Pi, B., Fu, S., Wang, W., & Han, S. (2009). SimHash-based Effective and Efficient Detecting of Near-Duplicate Short Messages.

[35] Ho T., Sung Kim., (2014). Fingerprint-Based Near-Duplicate Document Detection with Applications to SNS Spam Detection. International Journal of Distributed Sensor Networks. 2014. 8. 10.1155/2014/612970.

[36] TURAN, M , ÖGTELİK, S . (2018). İngilizce Dokümanlarda Tema ve Alt Kavramlar Tespit Modeli. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 6 (4), 754-764. DOI: 10.29130/dubited.420104