11th International Conference on Computer and Knowledge Engineering (ICCKE 2021), October 28-29, 2021, Ferdowsi University of Mashhad

Semi-automatic Detection of Persian Stopwords using FastText Library

Mohammad Dehghani*
Industrial and Systems Engineering
Tarbiat Modares University
Tehran, Iran
mohamad.dehqani@modares.ac.ir

Mohammad Manthouri
Electrical and Electronic Engineering Department
Shahed university
Tehran, Iran
mmanthouri@shahed.ac.ir

***Abstract*— A stopword is a word that does not add much semantic information to the text that despite of its very high frequency. Stopwords include prepositions, conjunctions, and pronouns. One of the steps in natural language processing is to remove stopwords to reduce dataset size and process faster. In this study, a semi-automatic method for collecting the Persian language's stopwords is proposed. The proposed method lists the stopwords of each text depending on its subject. For this purpose, based on a corpus of news texts, the Inverse Document Frequency (IDF) weight in the text is calculated for each word and the stopwords candidates are determined. Then, using the fastText library, the vector of each word is obtained. In the next step, five neighbors are found for each vector. Next, by removing duplicate words, the final list of stopwords (1014 stopwords) is collected. The result of simulations show the accuracy of detecting stopwords by the k-nearest neighbor method is 94.6%.**

***Keywords*— stopword; natural language processing; fastText; word embedding; text mining**

I. Introduction

One of the tasks in natural language processing is text preprocessing. On the one hand, the breadth of text information available on the Internet and the need to analyze this data, and on the other hand, the hardware and financial constraints of companies, lead researchers in this field to reduce the size of text data. A significant portion of text data in natural language processing and information retrieval are words that, despite their high frequency, do not contain much content information. Creating a list of stopwords can be very effective in reducing the size of text data. Stopwords are "high-frequency and non-significant words in a language that help to make sentences but do not add content to documents. Articles, prepositions, conjunctions, and pronouns are typical candidates [for removing stopwords] [1]. Creating a list of stopwords, in addition to being useful to researchers in the field of natural language processing and information retrieval, can also help business owners who need to analyze the information available on the Internet. For example, when analyzing official texts such as news or paper texts, it is possible to remove the stopwords. Since preprocessing is one of the most important steps in the analysis of text data, the existence of a comprehensive Persian list that is free and also, is useful for research in the field of natural language processing, is effective in reducing time and workload.

Researches related to the identification of stopwords can be divided into two general categories: non-automatic detection and automatic detection. In the non-automatic method, high-frequency words (often from one corpus) are collected and the expert decides whether to consider the word as a stopword. Lists that are collected in non-automatic methods cannot be used to work on every text. For instance, if the scientific or economic texts are to be examined, the stopwords of these subjects are sometimes specific and cannot be found in common lists. Also, social media data analysis has its own stopwords that are different from what is found in lists. In the automatic method, the identification and collection of stopwords are done with the help of machine learning or deep learning algorithms. In this article, a semi-automatic method for listing Persian language stopwords is introduced that can be used in natural language processing. The presented algorithm can list the stopwords of each text depending on its subject. The difference between the proposed method of this research and other researches is the use of word vectors and cosine similarity.

The purpose of word embedding is to present and store words in small vectors so that the vectors of synonyms have similar values. In the past, word embedding was based on statistical methods, but nowadays it is done with deep learning methods, such as the word2vec method, and has shown that semantic similarities between words can be extracted with its help. For example, embeds close to a word are usually synonyms. So, we can extract stopwords with the help of that. In other words, embeds store the conceptual information of words into a vector, and the neighbor embeds of a word are usually synonymous words, so the neighbors of a stopword are more likely to be considered as stopwords too.

FastText [2] is a free and public library that allows text representation. This library is prepared and usable for 157 languages. One of the main reasons for choosing this library

978-1-6654-0208-8/21/$31.00 ©2021 IEEE

for this research is the presence of a pre-trained model for the Persian language, which is not available in other libraries. Furthermore, to use other libraries such as word2vec, a large corpus of Persian texts has to be produced, and powerful hardware is needed, while the fastText library has used 600 billion words to build a model for the English language and created 2 million word vectors. In such cases, since it is not possible to collect Persian data in this size, a pre-trained model is used. The cosine similarity criterion is used to identify similar words; Which means that the similarity between two vectors is calculated from the inner product space. This formula calculates the cosine of the angle between two vectors and determines whether the two vectors are parallel or not [3]. This formula is also used in text similarity. In this research, cosine similarity is used to compare the similarity of each pair of words.

In this paper, a semi-automatic method for collecting the Persian language's stopwords is proposed. The stopwords of each text depending on its subject is introduced. In the first step, based on a corpus of news texts, the Inverse Document Frequency (IDF) weight in the text is calculated for each word and the stopwords candidates are determined. Secondly, using the fastText library, the vector of each word is obtained. In the third step, five neighbors are found for each vector. Next, by removing duplicate words, the final list of stopwords (1014 stopwords) is collected. The result of simulations show not only the method is semi-automatic but also the accuracy of detecting stopwords is significantly improving in comparison with other methods.

Firstly, we will review various researches in the field of stopword detection. In the following, the data used in the research are fully introduced. Then the proposed process for identifying stopwords is described step by step. In the end, the outputs are reviewed and compared with similar cases.

II. Related Work

Numerous studies have been conducted on the extraction of stopwords in different languages. One of the first researches in English is Fox [4]. Using Brown Corpus and the frequency of its words, as well as human supervision, Fox [4] was able to collect a list of English stopwords. Other researches have been done in different languages as well. As proof, Chen & Chen [5] and Hao & Hao [6] have extracted Chinese stopwords, the first of which is non-automatic and the second is automated using the weighted Chi-squared statistical criterion. Alajami et al. [7] have used statistical methods and [8] Alhadidi & Wedyan [8] have applied the combination of referring to the dictionary and using an algorithm to extract the stopwords of Arabic texts. Jha, V et al. [9] have also proposed an algorithm for extracting stopwords in Hindi which used the concept of Deterministic Finite Automata (DFM) . The difference of this research is there is no need to compare the text with a predefined list to determine the stopwords. Kumova Metin & Karaoğlan [10] have classified Turkish stopwords by discriminant analysis methods, decision tree, naïve bayes, and k-nearest neighbor and based on features such as Term Frequency (TF) , Collocative Frequency (CF), Document Frequency (DF), word's length, and word's position. Then this classification is completed by comparing the research data (Turkish) with the English data extracted from Brown Corpus. Rakib et al. [15] have worked on the stopwords of Bengali texts in two different ways. The innovation of this research is the use of Finite-state Automaton31 with 80% accuracy. Rani & Lobiyal, [16] have presented a list of Hindi language stopwords based on the vote ranking method. One of the advantages of this method is its ability to be used in a variety of texts with different structures. Also, removing the suggested list of this research significantly reduces the size of the text and increases the processing speed.

Researches have been done on Persian stopwords extraction, which can be divided into two categories: non-automatic and automatic extraction. Among the researches that have been done in the non-automatic approach, we can mention Taghva et al. [11] which, by exploring a collection of 1850 articles in online Persian newspapers and news, have proposed a list of stopwords. Davarpanah et al. [12] have also introduced stopwords based on linguistic and statistical criteria and expert’s judgment. As the authors of this article have pointed out, the stopwords constituted 39% of the text. One of the automatic researches on stopwords is Yaghoub-Zadeh-Fard et al. [13] that the aim is to provide a more accurate method while reducing the time and size of stored data, based on statistics and Part of Speech (POS) tags. Their presented method has a better performance compared to other methods such as entropy and document frequency and has a higher level of accuracy and retrieval compared to other proposed methods for recognizing Persian stopwords.

III. Method

The research data are collected from the texts of the Young Journalists Club website and then normalized with the Parsivar tool. Normalization includes tasks such as deleting non-Persian words and unauthorized characters, as well as correcting spelling mistakes. Selected topics are sports, world, politics, economy, social media, science & health, culture & art, and society. The number of selected news (documents) (by subject) can be seen in Table I and the details of the number of tokens can be seen in Table II.

TABLE I. Number of documents in each of the corpus topics

Topic	Number of documents
sports	15,963
world	19,782
politics	10,505
economy	13,835
social media	13,163
Science & health	9,594
culture & art	9,994
society	12,170
sports	15,963
world	19,782
politics	10,505
economy	13,835
total	165,091

TABLE II. Number of tokens in each of the corpus topics

Topic	Number of tokens
sports	5,000,061
world	5,000,012
politics	5,000,038
economy	5,000,291
social media	5,000,382
Science & health	4,999,611
culture & art	5,000,009
society	4,999,819
sports	5,000,061
world	5,000,012
politics	5,000,038
economy	5,000,291
total	60,000,625

Then the words are measured with the Inverse Document Frequency (IDF) 39 statistical criterion and with a minimum frequency limit of 800. The tokens that are obtained by applying this criterion and frequency limit are 3338 items, some of which can be seen in Table III. Equation (1) is used to obtain the IDF. In this formula, t is the term, d is the document, D is the total documents (in the corpus) and N is the total number of documents (in the corpus). Also, the phrase {d∈D: t∈d} refers to the number of documents in which the term t is has appeared.

$$Idf(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \quad (1)$$

Accordingly, the smaller the IDF number, the more frequently the term is used in the corpus. After calculation, each of the corpus terms is listed based on the IDF weight (ascending) as shown in Table III.

TABLE III: List of terms based on IDF minimum weight

Term	IDF Weight
باشگاه	1.015412
خبرنگاران	1.015790
به	1.016351
گروه	1.019317
در	1.030614
جوان	1.044247
از	1.082683
گزارش	1.130805
این	1.147860
با	1.155040
را	1.174103
است	1.180367

This list was reviewed by experts, and those which did not add content to the text were tagged as stopwords, and 313 items were selected. A list of these 313 primary stopwords is available in Appendix (end of article). Although the suggested method by the authors of this article has the feature of being used in different texts and topics, in this study, we tried to select only the most common stopwords according to the variety of topics. For this reason, based on the review of experts, some words that were not stopword at all (e.g., باشگاه or خبرنگاران), were removed from the candidates (313 stopwords). To resolve disputes between experts, a document was prepared to decide on all cases (71 cases). In this document, below each discussed term, reasons for rejecting or confirming the stopword are mentioned and specified in a separate column labeled N (not stopword) or Y (stopword). Such as, the verb بردن was one of the disputes, and it was finally decided not to be a stopword(N); Or شدن which was eventually considered a stopword(Y).

After identifying the stopwords by experts, each stopword was converted into a vector by the method of word embeddings and using the fastText library, and similar terms were obtained by the criterion of cosine similarity. The following step-by-step algorithm was defined and 313 previous step stopwords were presented in the form of a list named the "initial list" as input to the machine. Ultimately, after applying the above algorithm, for each stopword, five of the nearest neighbors were selected and added to the final list. The steps of the algorithm are as follows:

(0) Create an empty list called "final list";

(1) Select a term from the "initial list";

(2) Add the most similar terms(neighbors) to the selected term of the previous step in word embeddings (as regards the similarity of the two terms by cosine similarity) to the "final list";

(3) Select the next term from the "initial list" and go to step (2);

(4) Repeat steps(1) to (3) until the last term of "initial list";

(5) Remove duplicated terms from the "final list".

IV. Results Evaluation

The output of the algorithm, which is a table with 313 rows and 5 columns (313, 5), was reviewed by experts and non-stopword neighbors were labeled. Table IV shows some of the results of this analysis.

TABLE IV. Neighbors of each stopword

Term	First N.	Second N.	Third N.	Forth N.	Fifth N.
توان	نتوان	می‌توان	میتوان	بتوان	توان
آن‌ها	آن‌ها	خودشان	آن‌ها	آنان	آن‌ها

نبوده	نبود	نیست	نبوده‌اند	بوده	نبوده
عبارتند	قرارند	عبارتست	عبارتنداز	عبارت‌اند	عبارتند
چیست	کجاست	کدامند	چیه	چیست	چیست
نکته	نکته	نکته‌ی	نکته‌ای	نکات	نکته
آقای	آقاي	دکتر	اقای	جناب	آقای
دیگر	نیز	برخی	بسیاری	دیگری	دیگر
قطعا	یقینا	قطعاً	مسلما	مطمئنا	قطعا
حالا	اما	بعدش	الآن	الان	حالا

By adding the neighbors to the initial list, 1014 stopwords were eventually added. The following equation (2) was used to evaluate the percentage of precision in stopword detection. In this regard, tp is the number of true positives and fn is the number of false negatives. After calculating the precision for each of the columns, the result is shown in Table V:

$$Precision = tp/(tp+fn) \quad (2)$$

TABLE V. Precision of stopword detection

Number of columns	Number of errors (cumulative)	Precision(٪)
only the first column	8	97.4
the first and second columns	19	96.9
the first, second and third columns	34	96.3
the first to fourth columns	59	95.2
all columns	84	94.6

In this study, one of the challenges of data analysis was the duplication of neighbors. Case in point, the stopword «از» appeared in both «به» neighborhood (first neighbor) and «که» neighborhood (third neighbor). Duplicate terms were automatically removed when the final list was made.

To answer the question of whether the use of cosine similarity can be effective in finding stopwords that do not have specific semantic content or not, eight stopwords were randomly selected and the cosine similarity of each pair was calculated in Table VI. The number 1.0 is the highest and 0.1 is the lowest. To illustrate, the similarity of the two words «توانند» and «توانیم» is 0.576, which indicates the high similarity of these two words, but the similarity of the two words «توانند» and «آقای» is 0.111, which is a small number. Also, in the row related to the word «ایشان», the highest score belongs to the word «آقای», which seems logical. Based on the above explanations, it can be said that using the fasText library to identify stopwords, which do not have a specific meaning, has an acceptable performance. In other words, similar stopwords have similar vectors.

To evaluate the efficiency of the mentioned method, the accuracy obtained from this research has been compared with an article [10]. The reason for the comparison is the similarity of the methods of these two studies. The result is shown in Table VII. The criterion for comparison is the accuracy of the cosine similarity as well as the accuracy of the Euclidean distance of the terms.

TABLE VI. Cosine similarity of pairs of words

	اینکه	برخی	ایشان	حالا	توانند	توانیم	کی	آقای
اینکه	1.000	0.453	0.452	0.476	0.339	0.404	0.053	0.294
برخی	0.453	1.000	0.394	0.339	0.344	0.339	-0.022	0.157
ایشان	0.452	0.394	1.000	0.332	0.290	0.301	0.043	0.463
حالا	0.476	0.339	0.332	1.000	0.282	0.314	0.246	0.335
توانند	0.339	0.344	0.290	0.282	1.000	0.576	-0.003	0.111
توانیم	0.404	0.339	0.301	0.314	0.576	1.000	0.065	0.113
کی	0.053	-0.022	0.043	0.246	-0.003	0.065	1.000	0.083
آقای	0.294	0.157	0.463	0.335	0.111	0.113	0.083	1.000

TABLE VII. Comparison of the accuracy of word recognition

Criterion	Persian	English	Turkish
k- Nearest neighbor (Euclidean distance)	93.5	95.61	94.02
k- Nearest Neighbor (Cosine Distance)	94.6	95.11	93.41

V. Conclusion

In this research, firstly, the data of a Persian-language news site (Young Journalists Club website) were selected, and after the necessary pre-processing steps, the Inverse Document Frequency (IDF) was calculated for all the terms of the document. The corpus was selected from various news categories (are sports, world, politics, economy, social media, science & health, culture & art, and society), which were a total of more than 16,000 news (documents) and 60 million tokens. The terms were arranged in the ascending order based on IDF weight and after review by experts, 313 terms were selected as stopwords. In the examination of the candidates by experts, the terms that can be considered as stopwords in most contexts were selected. In the next step, an algorithm was applied using the fastText library on the news corpus and the vector of all the terms inside the corpus were extracted. Finally, with the help of cosine similarity, the five terms which had the most similarity to 313 stopwords were obtained. After analyzing the results, it was realized that this method has an accuracy of 94.6% in detecting stopwords.

The obtained accuracy is close to similar researches on Turkish and English languages, and the second method is better than the Turkish language research. According to Table VI and the investigations performed in the previous section, the model performs well in recognizing similar stopwords because, compared to traditional methods that predominately use bags of words, the model presented here takes into account semantic similarity of words in addition to the bag of words. By using the language model and the vector of each word, it is possible to store the summary of information about each word in a small vector. There are two advantages to this approach: Processing requires less memory and the meaning of each word is directly analyzed. Moreover, the model has the advantage of distinguishing verb-type stopwords.

As mentioned, the power of this algorithm is that it can be used in different subjects and with different texts. Automating the initial stopword selection process after calculating the IDF weight can be a suggestion for future work so that more machine intelligence can be perceived by diminishing the role of the expert.

A. Appendix

احتمالا اخیر اخیرا از ازای است
اصلا اغلب اکثر اکنون اگر اگرچه
اما الان انجام او ای ایشان
ایم این اینجا اینکه اینگونه آقای
آمد آمدن آمده آن آنان آنجا
آنچه آنقدر آنکه آنها آیا آید
با باتوجه باشد باشم باشند باشید
باشیم بالا بالاخره بالای باید بتوان
بتواند بتوانند بتوانیم البته بخواهد بخواهیم
بدست بدون بدهد بدهیم بدین بر
برابر براساس برای برخلاف برخی بطور
بعد بعضی بقیه بلافاصله بلکه بنابر
بنابراین بود بودم بودند بوده بودیم
بویژه بیاید بیایند بین پایین پس
پشت پیرامون پیش پیشتر پیشین تاکنون
تعدادی تقریبا تمامی تو توان تواند
توانست توانستند توانسته توانستیم توانم توانند
توانید توانیم جای جایی جز جلو
جلوی چرا چراکه چطور چگونه چنان
چنانچه چندان چندین چنین چون چه
چیزی چیست حالا حالی حالیکه حتما
حتی حداقل حداکثر حدود حدی حین
خاصی خواهد خواهم خواهند خواهید خواهیم
خود خودش خودم خیلی داد دادم
دادن دادند دادیم دارد دارم دارند
دارید داریم داشت داشتم داشتن داشتند
داشته داشتیم در درباره درحال درحالی
درخصوص دقیقا دوباره دهد دهند دهید
دهیم دیگر دیگران دیگری را رغم
روی زمانی زیر زیرا سایر سپس
سرانجام سوی شاید شد شدم شدن
شدند شده شدیم شما شود شوند
شوید شویم صرفا ضمن طرفی طور
طوری ظاهرا عبارتند عقب علاوه عملا
غیرقابل غیره فعلا فقط قبل قبلا
قبلی قبیل قطعا کاملا کجا کدام
کرد کردم کردن کردند کرده کردید
کردیم کسی کلی کمتر کمی کنار
کند کنم کنند کنونی کنید کنیم
که کی گاه گاهی گردد گردید
لحاظ لذا ما مانند مثل مثلا
مدنظر مرا مرتبط مقابل مگر من
منظور می میان ناگهان ناگهانی نبوده
نحوه نخواهد نشود نکته نکرد نکرده
نکنند نکنید نکنیم نماید نوعی نیز
نیست نیستم واقعا وقتی ولی های
هایش هایشان هر هرچند هرگونه هست
هستند هستید هستیم هم همان همانطور
همانند همچنان همچنین همچون همزمان همگی
همواره همه همیشه همین هنگام هنگامی
هنوز هیچگاه هیچگونه الی یا یابد
یابند یافت یافتن یعنی یک یکدیگر
یکی

REFERENCES

[1] Liu, B., Web data mining: Exploring hyperlinks, contents, and usage data (2nd ed). Springer, 2011.

[2] K.-H. Chen and H.-H. Chen, “Cross-language Chinese text retrieval in NTCIR workshop: towards cross-language multilingual text retrieval,” in ACM SIGIR Forum, 2001, vol. 35, no. 2, pp. 12-19: ACM New York, NY, USA.

https://doi.org/10.1162/tacl_a_00051

[3] Han, J., Kamber, M., & Pei, J., Data mining: Concepts and techniques (3rd ed). Elsevier, 2012.

[4] Fox, C., “A Stop List for General Text”, SIGIR Forum, Vol. 24, No. 1–2, pp. 19–21, 1989,

https://doi.org/10.1145/378881.378888.

[5] Chen, K., & Chen, H.-H., “Cross-Language Chinese Text Retrieval” in NTCIR Workshop: Towards Cross-Language Multilingual Text Retrieval”, SIGIR Forum, Vol. 35, No. 2, pp. 12–19, 2001, https://doi.org/10.1145/511144.511149.

[6] Hao, L., & Hao, L., “Automatic Identification of Stop Words in Chinese Text Classification”, 2008 International Conference on Computer Science and Software Engineering, pp. 718–722, 2008, https://doi.org/10.1109/CSSE.2008.829.

[7] Alajmi, A., Saad, E., & Darwish, R. R., “Toward an ARABIC Stop-Words List Generation”, International Journal of Computer Applications, Vol. 46, pp. 8–13, 2012.

[8] Alhadidi, B., & Wedyan, M, “Hybrid Stop-Word Removal Technique for Arabic Language”, Egyptian Computer Science Journal, Vol. 30, pp. 35–38, 2008.

[9] Jha, V., Manjunath, N., Shenoy, P. D., & Venugopal, K. R., “HSRA: Hindi stopword removal algorithm”, 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), pp. 1–5, 2016, https://doi.org/10.1109/MicroCom.2016.7522593.

[10] Kumova Metin, S., & Karaoğlan, B., , “Stop Word Detection as A Binary Classification Problem,”, Anadolu University Journal of Science and Technology A - Applied Sciences and Engineering, Vol. 18, pp. 346-359, 2017, 10.18038/aubtda.322136.

[11] Taghva, K, Beckley, R, Sadeh, M., “A list of farsi stopwords”, Technical Report 2003-01. Information Science Research Institute, University of Nevada, Las Vegas, NV, 2003.

[12] Davarpanah, M. R., Sanji, M., & Aramideh, M., “Farsi lexical analysis and stop word list”, Library Hi Tech, Vol. 27, No. 3, pp. 435–449, 2009, https://doi.org/10.1108/07378830910988559.

[13] Yaghoub-Zadeh-Fard, M.-A., Minaei-Bidgoli, B., Rahmani, S., & Shahrivari, S., “PSWG: An automatic stop-word list generator for Persian information retrieval systems based on similarity function & POS information”, 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 111–117, 2015 https://doi.org/10.1109/KBEI.2015.7436031

[14] Mohtaj, S., Roshanfekr, B., Zafarian, A., & Asghari, H., Parsivar: “A Language Processing Toolkit for Persian”, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018 https://www.aclweb.org/anthology/L18-1179

[15] R. U. Haque, M. F. Mridha, M. A. Hamid, M. Abdullah-Al-Wadud, and M. S. Islam, “Bengali stop word and phrase detection mechanism,” Arab. J. Sci. Eng., vol. 45, no. 4, pp. 3355–3368, 2020.

https://doi.org/10.1007/s13369-020-04388-8

[16] R. Rani and D. K. Lobiyal, “Performance evaluation of text-mining models with Hindi stopwords lists,” J. King Saud Univ. - Comput. Inf. Sci.,2020.