International Journal of Information Science and Management
Vol. 23, No. 1, 2025, 91-107
DOI: https://doi.org/10.22034/ijism.2025.2017335.1322 / DOR: https://dor.org/20.1001.1.20088302.2025.23.1.7.1

Original Research

Analytical Comparison of Stop Word Recognition Methods in Persian Texts

Mohammad Ebrahim Samie
Assistant Prof., Department of Computer Engineering and IT, Jahrom University, Jahrom, Iran.
Corresponding Author: samie@jahromu.ac.ir
ORCID iD: https://orcid.org/0000-0002-7109-2980

Erta Bahmani
BSc, Department of Computer Engineering and IT, Jahrom University, Jahrom, Iran.
ertabhmni@gmail.com
ORCID iD: https://orcid.org/0000-0002-9882-239X

Niloofar Mozafari
Assistant Prof., Islamic World Science and Technology Monitoring and Citation Institute (ISC), Shiraz, Iran.
mozafari@isc.ac.ir
ORCID iD: https://orcid.org/0000-0002-8129-9918

Received: 07 December 2023
Accepted: 03 July 2024

Abstract

Stop words are primarily non-significant words used to connect other words in sentence construction. Since these words do not contain specific information about the text, they are typically removed during text processing. Therefore, identifying stop words is an essential operation in text processing. A challenge arises when usually insignificant words can become significant depending on the situation, while words that are typically important can sometimes be classified as stop words. This problem is particularly pronounced in Persian due to the complexities inherent in the language. Recognizing the importance of identifying stop words in Persian, we analyzed and reviewed various approaches, including a dictionary-based approach, POS tagging-based approach, Word2Vec-based approach and FastText-based approach to identify stop words using a corpus of 50.000 Persian sentences from Hamshahri dataset. Our findings indicate that the FastText-based approach outperformed the others with a detection accuracy of 96.98, suggesting that this method can lead to the development of an automatic, reliable, and efficient system.

Keywords: Stop Words, Content Words, Persian Language Processing, POS Tagging, Word2Vec, Fasttext.

Introduction

In natural language processing, text construction utilizes two types of words: stop words and content words. Despite their frequency, stop words have minimal effect on the overall meaning, and they are often excluded with minimal impact on the sentence. Stop word detection is a text-preprocessing technique widely used in sentiment analysis (Chanda & Pal, 2023; Jefriyanto, Ainun & Al Ardha, 2023). Conversely, content words carry significant semantic weight; their removal can entirely alter the sentence's meaning (Namely, Bouzoubaa & Yousfi,

2019). In many text-processing tasks, content words are considered pivotal. Stop words are often scattered throughout documents, while their presence in sentences has the least semantic impact, so they should be removed from any document to improve language description (Chanda & Pal, 2023). In natural language processing, identifying and removing stop words is essential in increasing the efficiency and accuracy of text analysis.

The complexity and subtlety of Persian present a considerable challenge when recognizing stop words. Unlike English, where stop words like prepositions and conjunctions are easy to identify due to their structure and syntax, in Persian, words typically considered stop words in English may carry substantial meaning, depending on the context and their position within a sentence. For example, "the" in English is a stop word that cannot be content in any context and text. Still, there are many words in the Persian language that, depending on the meaning of the sentence and the place of the word in the sentence, can sometimes be a stop word and sometimes a content word.

In Persian, prepositions are not meaningful on their own. They gain meaning by being placed next to other words, typically before nouns, pronouns, and noun phrases. Let's explore this with an example: the words "be" and "rɒ" are prepositions in Persian and are considered stop words by themselves. In the sentence “mes'æle ri'æzi rɒ be 'hobi 'hæl kor'dæm” (I solved the mathematical problem well), the words "be" and "rɒ" are stop words. Now consider the sentence: “mes'æle ri'æzi 'hobi 'hæl kor'dæm” (literally: I solved a good mathematical problem). By removing these two words, the sentence loses its meaning and becomes different. There are many examples that illustrate the complexity of the Persian language. For instance, in the sentence "tærz tehiː'e sæ'hiːh suːp 'dʒuː tʃiːst?” (literally: What is the correct way to prepare barley soup?), the word "sæ'hiːh" (correct), which is generally a content word, becomes a stop word. Its removal does not change the meaning of the sentence.

Considering the importance of identifying stop words in Persian, the researchers attempted to fill this research gap by using Persian stop words’ distinct challenges—this effort Persian language processing methods and textual analysis. Using the latest approaches in the field of natural language processing, including the dictionary method with TF-IDF vectorizer, POS tagging, Word2Vec, and FastText, the researchers attempted to devise a method to identify stop words in Persian. The study relied on Hamshahri's extensive corpus (AleAhmad, Amiri, Darrudi, Rahgozar, & Oroumchian, 2009) comprising 50,000 Persian sentences in different news and fiction fields for data analysis.

In the next section, we will review stop word recognition methods. In the third part, we will explain the research method. Next, in the fourth part, we will analyze the results and conclude the work.

Literature Review

In text analysis, efforts have been made to identify and exclude stop words. The concept of "stop word" originated from Luhn’s (1985) pioneering work titled "Keyword-in-context index for technical literature (KWIC Index)," which provided a fundamental tool for indexing. Generally, the KWIC index distinguishes between "keywords" and "non-significant" words. "Keywords" are basic terms or phrases that indicate the main content and meaning of a document. They are identified using a specialized code based on specific rules. These words matter in the sentence and are essential in helping users find the most relevant information from the document. On the contrary, "non-significant" words include everyday language elements such as conjunctions, prepositions, auxiliary verbs and unique adjectives that may be

grammatically essential but typically hold little value for information retrieval, as they do not carry the text's primary meaning.

Hao and Hao (2008) identified a list of 500 Chinese stop words by applying Chi-squared statistical weighting to words in Chinese texts, markedly improving the micro F1 score. Davarpanah, Sanji, and Aramideh (2009) presented a comprehensive method for constructing a list of stop words in Persian, considering syntactic classification, domain dependence, statistical data, and expert opinions. The results obtained from these techniques culminated in a list of 927 Persian stop words. In addition, the researchers examined some of the main challenges in the automatic processing of Persian texts to improve Persian text processing methods (ibid). Examining statistical methods to identify Mongolian stop words in Mongolian information retrieval, Mongolian researchers combined two models, calculating entropy and part of speech, to remove nouns, verbs, or homonyms (words with the same pronunciation but different meanings), thus refining information retrieval in Mongolian texts (Zheng & Gaowa, 2010).

In a 2012 study, a method was developed for stop word detection from readability assessment (i.e., the level of ease of reading and comprehension of a text) in Thai text using both statistical and informational models for de-compilation. The study showed that the F-measure obtained by removing stop words using the Support Vector Machine (SVM) algorithm was 87% for medium grades. Still, it achieved poor results for lower grades by removing stop words (Daowadung & Chen, 2012).

Using a statistical strategy for building an Arabic stop word list, Arab researchers developed a list of Arabic stop words, derived from the general stop word list using an artificial neural network (ANN) classifier, with an efficiency of 96% (Alajmi, Saad & Darwish, 2012).

Iranian researchers used numerical data to measure the frequency and importance of words (Sadeghi & Vegas, 2014). They proposed a method based on a periodicity-based automatic aggregation, inverse document periodic normalization, and an information retrieval model to extract light stop words from Persian texts. The evaluation results indicated a significant alignment between Persian and English stop words with a substantial improvement in reducing the index term. Specifically, 32 Persian stop words have a significant effect in reducing the size of the index, and the set of stop words can reduce the number of index terms by approximately 27%.

In a 2014 study, a new algorithm was developed for structurally removing stop words similar to root words. The algorithm’s effectiveness was evaluated using simple Bayesian methods for the Kannada language. The dataset for the study was constructed using Technology Development for Indian Languages (TDIL) (Jayashree, Murthy & Anami, 2014). In the same year, a method was developed to detect and remove stop words from Twitter texts automatically. This method involved semantic and sentiment analysis using the Senti Circle model. The latter created a binary sentiment classifier with 94% accuracy. Unlike traditional methods that rely on predefined lists of stop words, this method improved the feature space and distribution of the data (Saif, Fernandez & Alani, 2014). Wilbur and Sirotkin (1992) used a different document-document similarity approach to identify and remove stop words from the MEDLINE database.

Yaghoub-Zadeh-Fard, Minaei-Bidgoli, Rahmani, and Shahrivari (2015) introduced a comprehensive method for automatically generating stop word lists for Persian information retrieval systems. The method included automatically generating stop word lists and extracting

Persian words based on POS tags and BM25 ranking, which led to the identification of low-scoring words as stop words.

A 2016 study attempted to create a list of Malay stop words from the Dewan Bahasa Dan Pustaka (DBP) corpus by combining statistical methods, distribution methods, and entropy measurement. The study produced a list of 339 Malay stop words (Chekima & Alfred, 2016). Shaukat et al. also used a deterministic finite element (DFA) to remove stop words from Urdu documents (Dar, Shafat & Hassan, 2017).

Metïn and Karaoğlan (2017) showed that stop words are successfully recognizable in Turkish texts using a binary classification method. This method performed comparably to the frequency-based methods. However, it was not as successful in an English corpus.

In a 2018 study, a method utilized finite deterministic automata to detect and remove English stop words in English text retrieval. Achieving 99% accuracy, the stop word removal process took only 1.78 seconds to run, significantly outperforming pattern-matching techniques (Behera, 2018). In the same year, researchers introduced a new method for generating domain-specific stop word lists from Sinhala newspapers using Naïve Bayes and Max Entropy models. The results indicated an accuracy of 99.74% in normalized inverse document frequency (NIDF) without stop words. Max Entropy showed more sensitivity than Naïve Bayes to remove stop words (Gunasekara & Haddela, 2018). Choi, Yoo, and Choi (2019) extracted Korean stop words and typos using the weighted TF-IDF method. Miretie and Khedkar (2018) also indexed stop words from Amharic texts using TF, IDF, entropy criteria.

In a 2019 study, a comprehensive mechanism was devised for detecting stop words in Bengali using corpus-based pattern matching, which led to improvements in detecting and removing stop words. This method achieved 70-75% accuracy using pattern-matching techniques (ul Haque, Mehera, Mridha & Hamid, 2019).

In a 2020 study, the researchers combined corpus-based and finite-state automatic methods to classify and identify Bengali stop words and phrases. The corpus-based method performed better than the limited-mode automatic approach in both stop word and phrase recognition (Haque, Mridha, Hamid, Abdullah-Al-Wadud & Islam, 2020).In the same year, the Simple Bayesian method and the Complementary Simple Bayesian approach were found to be effective for the removal of stop words using frequency-based methods and lexical resources in Tamil texts (Rajkumar, Subashini, Rajan & Ramalingam, 2020).

In a 2021 study, a semi-automated method was utilized for compiling Persian keywords (Dehghani & Manthouri, 2021). The proposed method classified the keywords based on the topic of the text. Using the k-nearest neighbor (k-NN) approach, the method achieved a recognition accuracy of 94.6%.

In a 2022 study, researchers aimed to automatically analyze and recognize stop words in Uzbek texts, considering the challenges of the language's agglutinative nature. Using a newly prepared corpus, they applied the bigram method and devised a more effective compositional method for stop word recognition. Focusing on 731,156 Uzbek words in the "School Corpus", the researchers highlighted the complexity of stop word recognition in affix languages compared to agglutinative ones. The results showed that the harmony method is six times more efficient than the bigram method in identifying end words in Uzbek texts (Madatov, Bekchanov & Vičič, 2022).

A recent study investigated the effect of pre-processing on word embeddings in Persian and English sets, with special emphasis on changing the number of co-occurrences through pre-

processing approaches such as removing stop words (Rahimi & Homayounpour, 2023). The study found that eliminating punctuation marks and keywords can increase performance depending on the task, especially in text classification and semantic comparison of words. Conversely, for sentiment analysis and syntactic word analogy, retention of stop words was found to lead to improved overall performance.

After reviewing many research papers in different languages e.g., Bengali, Gujarati, Persian, English, Sinhalese, Arabic, Kannada, Tamil, Thai, Malay, Urdu, Chinese, Turkish, Mongolian, Korean, Amharic etc., we were convinced that addressing this specific need to improve stop word lists in way that is appropriate for Persian texts would contribute to the evolving field of Persian language processing. While previous research has made significant progress in this area, our study investigated different methods for identifying stop words on a 50.000 sentence corpus from the extensive Hamshahri corpus; this comprehensive analysis allowed us to determine stop words to identify and compare different areas and finally reach valuable insights in this area.

Materials and Methods

This section explains the data used, data pre-processing, and the methods used to compare the results.

Dataset

We used a 50,000-sentence corpus called Hamshahri, a large Persian corpus based on the Iranian newspaper Hamshahri. After the validation and verification process, we prepared a data set for further execution and divided it into 80 and 20 for training and testing.

In this data set (Figure 1), the most common word is "and" with the highest frequency (67054).

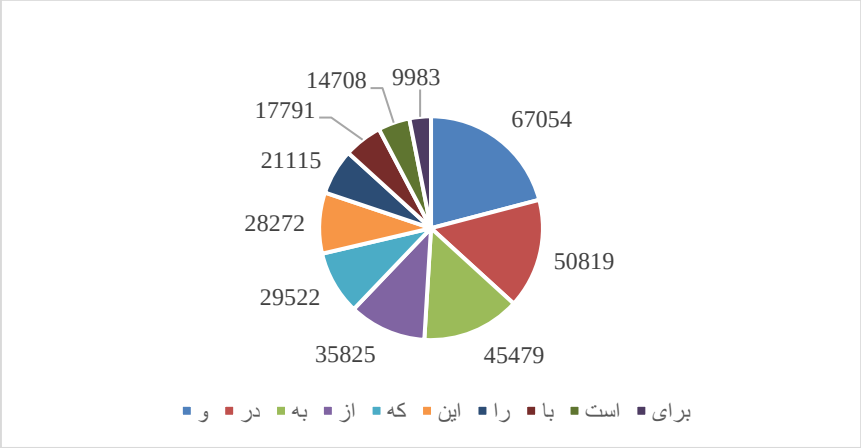

Figure 1: Ten most frequent words in the dataset

Table 1 provides a detailed analysis of the environment and dataset used in this study. It enumerates the following factors: the library utilized, the programming language in which the study was conducted, and the development of the environment for the analysis. Additionally, it presents key metrics of the dataset, including the total number of sentences, the count of sentences with stop words, and those without stop words. The table also specifies the maximum sentence length, the total number of words, and the total number of stop words found in the

dataset. This comprehensive breakdown offers an in-depth understanding of the structure of the dataset and the tools used for analysis.

Table 1
Analysis of environment and dataset

Library	Scikit-Learn, NLTK, hazm, genism, Numpy, StanfordNlp
Language	Python
Development Environment	Jupyter-Notebook
Total No. of Sentences	50,000
Sentences with Stop Words	49,384
Sentences Without Stop Words	616
Maximum Sentence Length	448
Total Words	1425630
Total Stop Words	1371

Dataset preprocessing

The implementation dataset was pre-processed, so we first identified the sentences and then marked them to boost performance.

1) During the data preprocessing phase, all punctuation marks, numerical figures, symbols, and irrelevant information in Table 2 were eliminated to facilitate processing efficiency.

Table 2
Examples of useless information

A	B	C	D	E	F	G	H	I	J	K
L	M	N	O	P	Q	R	S	T	U	V
W	X	Y	Z	A	b	c	d	e	f	g
h	i	j	k	L	m	n	o	p	q	r
s	t	u	v	W	x	y	z	0	1	2
3	4	5	6	7	8	9	!	@	#	$
%	^	&	*	(	)	-	-	=	+	\
|	/	?	[	]	{	}	,	<	>	.
‘	0	1	2	3	4	5	6	7	8	9
÷	×	،	؛	...	:	>>	<<	.	"	؟

2) Segmentation of sentences: Each sentence's word was considered an attribute, which was subsequently divided into characters. Each feature of the sentence was labeled according to its position by checking the index value.

In Table 3, Check for index "1" is one; in the sense that the word "'xili:" (literally: many) many is a stop word, also without this word, there is no change in the actual meaning of the sentence. The assigned value is zero for index "0", "2" and "3" because these words are content.
Table 3

An example of a sentence in the data set: " man 'xili: fær'zænd 'dɒræm" (literally: I have many children)

Check	Position	3	2	1	0
0	0	دارم	فرزند	خیلی	من
1	1	دارم	فرزند	خیلی	من
0	2	دارم	فرزند	خیلی	من
0	3	دارم	فرزند	خیلی	من

Dictionary method

We built a dictionary with stop words for the dictionary-based approach with TF-IDF vectorizer sequentially to convert the raw sentences into numeric values. To do that, the numerical values for each sentence were generated using a TF-IDF vectorizer. The TF-IDF vectorizer was applied to the sentences with a dictionary containing stop words. The feature vector was constructed using the TF-IDF vectorizer and then added to the data. A corresponding vector was created for each sentence using the dictionary, incorporating the stop words. This resulted in each sentence having a numerical feature representing its characteristics.

Table 4 shows the final unique values of words for the sentence 'man nmi'xɒhæm be'rævæm' (literally: I don’t want to go).

Table 4

The final unique value of words for the sentence " man nmi'xɒhæm be'rævæm" (literally: I don’t want to go)

بروم	نمی‌خواهم	من
0.70710678	0.70710678	0

Figure 2 illustrates the accuracy of four classifiers-Logistic Regression, Support Vector Machine (SVM), Random Forest, and Decision Using the dictionary method with a TF-IDF vectorizer for detecting Persian stop words. Each performance of each classifier was averaged over 10 iterations. Random Forest demonstrated the highest accuracy at 99.04%, effectively capturing complex patterns with its ensemble method. Logistic Regression achieved a slightly lower accuracy at 98.83%, performing well but not quite matching the performance of Random Forest. Support Vector Machine (SVM) performed well but below Random Forest and Logistic Regression, while Decision Tree had the lowest accuracy, likely due to overfitting in high-dimensional TF-IDF spaces. In conclusion, Random Forest emerged as the best classifier with the highest accuracy, making it the most reliable choice for this task.

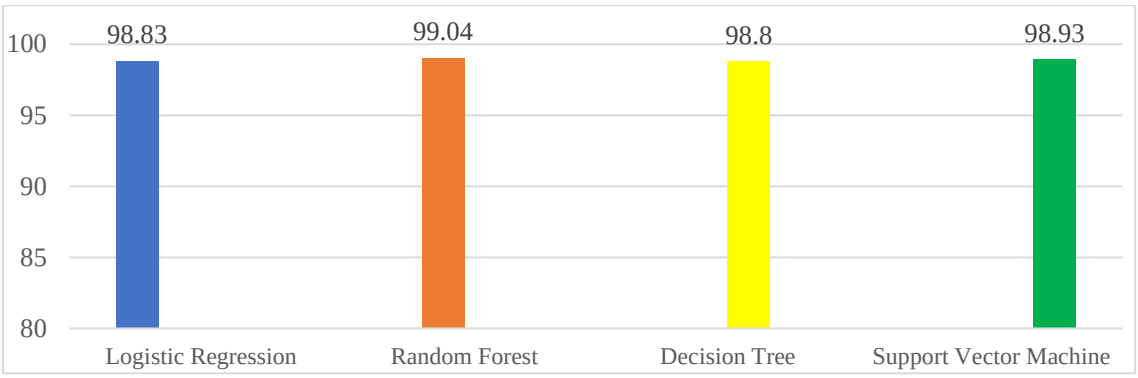

Figure 2: Accuracy of dictionary method with TF-IDF vectorizer

POS tagging method

POS tagging is a natural language processing technique that assigns specific grammatical categories or parts of speech (POS) to each word in a sentence or text. Some categories in the Persian language are nouns, subjects, objects, adverbs, pronouns, verbs, prepositions, conjunctions, etc. Using the StanfordNLP library, we used a powerful tool to perform part-of-speech tagging on Persian text. The library's capabilities allowed us to tag each word according to its Persian position in a sentence. StanfordNLP uses advanced techniques, including dependency decomposition and second-order Markov models, to accurately predict each word's grammatical category (Manning, Surdeanu, Bauer, Finkel, Bethard & McClosky, 2014).

Table 5 shows the dataset tagged with POS after numerical transformation. In this table, the value "0" indicates the end of a meaningful sequence of POS tags. It acts as a terminator, signifying no further relevant POS tags after this point. This helps differentiate between actual POS tags and padding values. Furthermore, in the "Check" section of the encoding scheme, the value "0" denotes a stop word, while the value "1" represents a content word. This distinction aids in identifying stop words and content words within the text data during the processing and analysis stages.

Table 5

Dataset tagged with POS after numerical transformation

No.	Texts	Position	Check	POS_list	Encoded_POS_list
0	من نمیخواهم بروم	0	0	[PRON, VERB, VERB]	[11,15,15,0, 0,…]
1	من نمیخواهم بروم	1	1	[PRON, VERB, VERB]	[11,15,15,0, 0,…]
2	من نمیخواهم بروم	2	1	[PRON, VERB, VERB]	[11,15,15,0, 0,…]

Figure 3 illustrates the accuracy of various POS tagging approaches. The Support Vector Machine (SVM) method, outperforming the other techniques evaluated, demonstrates the highest accuracy. This superior performance can be attributed to SVM's ability to handle high-dimensional data and create optimal hyperplanes for classification, effectively distinguishing

between different parts of speech. Conversely, the decision tree approach exhibits the lowest accuracy, likely due to its tendency to overfit, especially in high-dimensional spaces created by the features used for POS tagging.

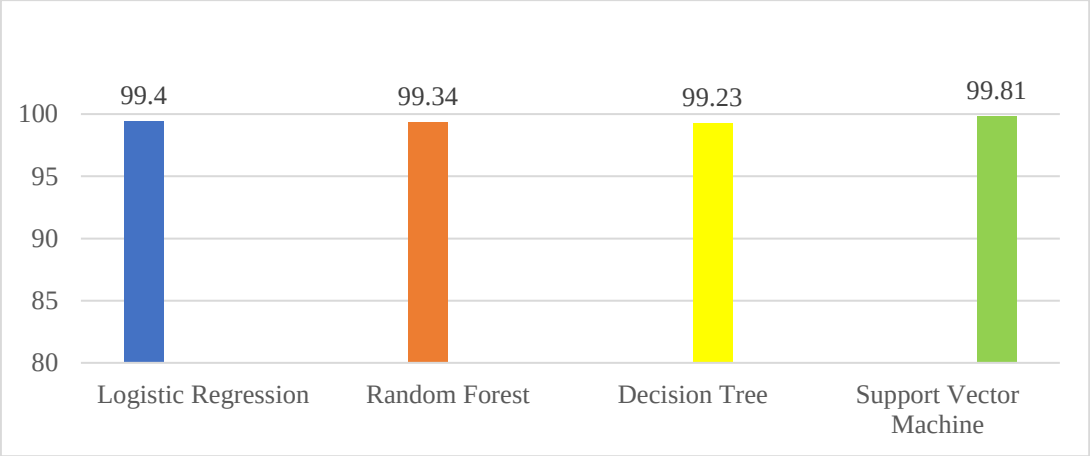

Figure 3: Accuracy of POS tagging approach

Fast text-based method (Word2Vec)

Word2Vec, and FastText highlight word embedding techniques were used to recognize stop words from sentences to enrich the research source of Persian language processing. Word2Vec groups corresponding words in a text to help distinguish informative from non-informative words. It creates a dimensionally defined vector space from the input, assigns a corresponding vector to each unique word in the dataset, and creates a computationally efficient predictive model for learning word embeddings (Mikolov, Chen, Corrado & Dean, 2013).

Table 6 elucidates the process of sentence segmentation, delineating the binary classification of tokens as either stop words, denoted by a value of 0, or content words, represented by a value of 1. Thus, the segmentation within the corpus is elucidated.

Table 6

The sentence is tokenized in the WORD2VEC model

No.	Texts	Position	Check	Tokenized_sentences
0	من نمیخواهم بروم	0	0	[من ,نمیخواهم ,بروم]
1	من نمیخواهم بروم	1	1	[من ,نمیخواهم ,بروم]
2	من نمیخواهم بروم	2	1	[من ,نمیخواهم ,بروم]

Table 7 presents word vectors obtained using the Word2Vec model, a robust algorithm for generating word embeddings trained on a vast corpus of text data. Each row in Table 6 corresponds to a word vector, where words such as 'من' (I), 'نمیخواهم' (do not want), and 'بروم' (go) are represented in a high-dimensional space capturing semantic relationships. These word vectors encode semantic features, enabling analysis of semantic similarities and relationships between words. The table illustrates the semantic context and associations learned by the Word2Vec model, providing valuable insight into the underlying structure of the language.

Table 7
Vector representation of words using the WORD2VEC model

No.	W2 v_0	W2 v_1	W2 v_2	W2 v_3	W2 v_4	…	W2v_298	W2v_299
0	1.267079	1.751248	0.444342	1.10324	-0.974945	…	-0.194166	0.279045
1	-0.021621	0.137409	0.011457	0.132269	0.001083	…	0.05778	-0.019896
2	-0.107528	0.232795	-0.238764	0.467439	-0.267386	…	0.142185	0.021192

Fast Text is a library created by the Facebook research team as an extension of Word2Vec to learn word representation and sentence classification more efficiently. It can be multi-processed during training and train supervised and unsupervised word and sentence representations. FastText considers each word to consist of n-gram characters; a word vector is the sum of these n-gram characters, while Word2Vec considers each word an atomic entity. For example, the vector word "beauty" is the sum of n-gram vectors.

"<be", "bea", "beau", "beaut", "beauty", "beauty>", "eau", "eaut", "eauty", "eauty>", "aut", "auty”,” auty>”, “uty”,” uty>”,” ty>”.

The technique utilizes our data and trains the model with the Continuous-Bag-of-Words (CBOW) FastText function. Figure 4 illustrates the accuracy of the Word2Vec, explicitly using the Continuous-Bag-of-Words (CBOW) model. This figure showcases the performance metrics and effectiveness of the CBOW method in accurately capturing the contextual relationships between words in the dataset. The Continuous-Bag-of-Words model combines all textual word vectors to represent the target word. It utilizes the Word2Vec CBOW model, known for its robustness, achieving a 99.53% accuracy rate in Decision Tree and Logistic Regression classification, demonstrating superior performance.

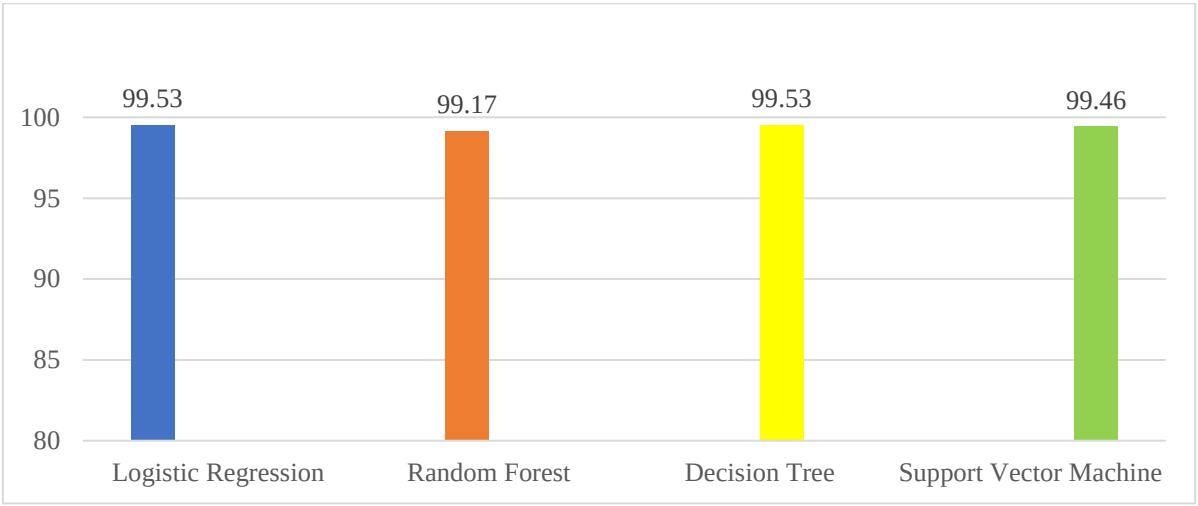

Figure 4: Accuracy of Word2Vec approach based on Continuous-Bag-of-Words (CBOW)

Table 8 presents the results of applying the FastText algorithm with specific parameters for generating word embeddings. The 'vector_size' parameter sets the dimensionality of word vectors to 300, capturing nuanced semantic information. A 'window' size of 5 considers contextual words within a sentence. Character 5-grams, defined by 'min_n=5' and 'max_n=5',

are utilized for training. Additionally, 'negative=10' specifies the number of negative samples used during training for improved model performance.

Table 8
Parameters used in pre-trained Persian FASTTEXT model

Model	CBOW
Dimension	300
Length of character n-gram	5
Window size	5
Negatives	10
Max_n	5
Min_n	5

Figure 5 illustrates the focus on the accuracy of the Continuous-bag-of-words (CBOW) based FastText. The results demonstrate varying performances across different classification algorithms. Incorporating the Skip-Gram model, which utilizes the current word to forecast word vectors in various positions, enhances the model's sensitivity to word placements within the text. Specifically, the support vector machine model exhibited an impressive accuracy of 99.47%. This outcome underscores the model's effectiveness in text classification tasks and its robust capabilities in managing intricate text nuances.

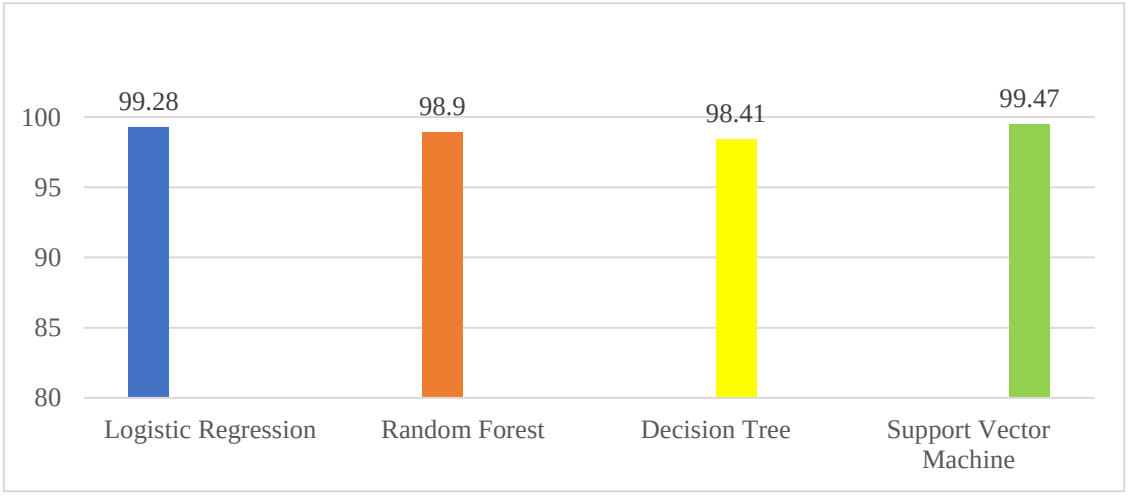

Figure 5: Accuracy of FastText approach based on Continuous-Bag-of-Words (CBOW)

Table 9 summarizes the parameters used in configuring the WORD2VEX SKIP-GRAM model. The embedding size is set to 300, indicating that a 300-dimensional vector will represent each word. A context window size of 4 means that the model considers four words to the left and right of the target word for the context. The `sag` parameter is set to 1, specifying using the Skip-gram architecture. The `has` parameter is set to 0, indicating that hierarchical softmax is not used. The `min_count` parameter is set to 1, ensuring that all words in the corpus are included in the training, regardless of their frequency. The model utilizes four worker threads to parallel the training process. Negative sampling is disabled (`negative` is 0), and the model is trained for 10 iterations to optimize the word vectors.

Table 9
Parameters used in the WORD2VEC SKIP-GRAM model

Parameters	Value
size	300
window size	4
sg	1
hs	0
min_count	1
workers	4
negative	0
iteration	10

Figure 6 evaluates classifiers employing the Word2Vec skip-gram technique to detect Persian stop words. The classifiers assessed encompass Logistic Regression, Random Forest, Decision Tree, and Support Vector Machine, achieving accuracies of approximately 99.34%, 99.27%, 99.61%, and 99.34%, respectively. These findings underscore the efficacy of the Word2Vec skip-gram method in discerning Persian stop words from content words, with the Random Forest classifier exhibiting the highest accuracy rate among the models evaluated.

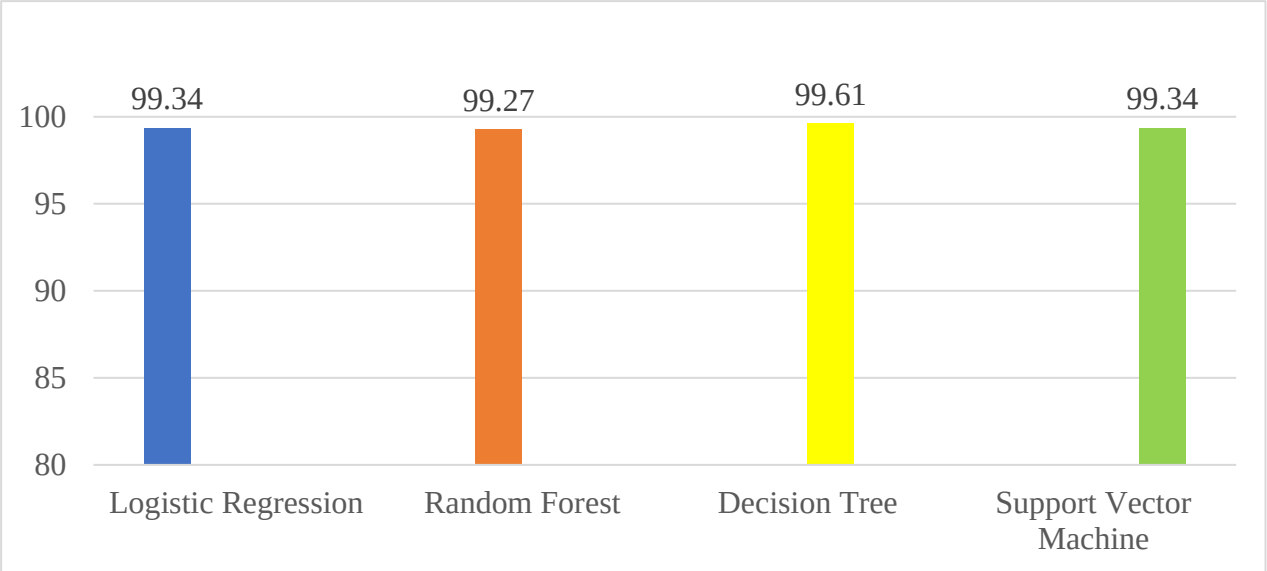

Figure 6: word2vec skip-gram based method

Discussion

After extensively investigating various stop-word detection methods in Persian texts, we concluded that integrating Part-of-Speech (POS) tagging with logistic regression modeling is the most effective and efficient approach for detecting stop words in Persian. This result is also consistent with the results of some previous studies. For instance, Chiche and Yitagesu (2022) provided a comprehensive review of advancements in POS tagging, emphasizing challenges

like ambiguity and unknown words while exploring solutions using Machine Learning (ML) and Deep Learning (DL) techniques. Another study explored logistic regression, along with random forest and K-nearest neighbors (KNN), to classify BBC news articles and highlight logistic regression as the most effective algorithm for text classification. These studies show the power of these two algorithms, POS tagging and logistic regression. Raja, Tasharofi, and Oroumchian (2007) also achieved high accuracy using POS tagging in conjunction with a machine learning algorithm, aligning with the effectiveness of combining linguistic features with machine learning methods for Persian stop word detection.

Integrating POS tagging with logistic regression yielded highly effective results, demonstrating a robust and accurate approach for stop word detection in Persian texts. However, by combining these two methods, this study achieved even better results, enhancing the accuracy and efficiency of stop-word detection. Kholwal (2023) also declares that this integration is compelling. This synergy led to a remarkable accuracy of 91.98% and showed considerable power in addressing the complex grammatical structures and subtle linguistic differences inherent in the Persian language.

As shown in Figure 7, the POS tagging method, which emphasizes the use of grammatical context and linguistic features, demonstrated its superiority in accurately classifying words in Persian sentences (Chiche & Yitagesu, 2022). By using part-of-speech information, this approach successfully discriminated between stop words and content words, resulting in accurate and reliable stop word detection. Similar approaches in other studies have shown that part-of-speech tagging is crucial in distinguishing stop words from content words (Dehghani & Manthouri, 2021).

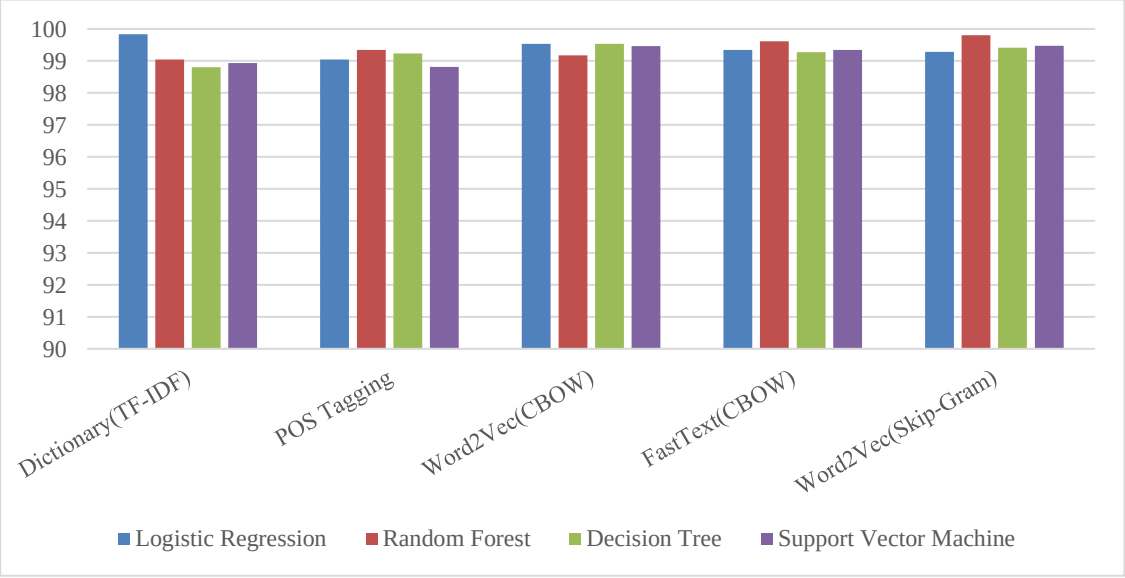

Figure 7: Performance analysis of different approaches with supervised machine learning classifiers

Compared to the alternative methods, such as the dictionary-based approach, Word2Vec, and FastText models, the POS tagging method combined with logistic regression outperformed in identifying Persian stop words. Some other studies also confirm these comparisons, such as those in the (Kholwal, 2023) (Raulji & Saini, 2016), (Metïn & Karaoğlan, 2017). The ability to

combine linguistic context with grammatical insights effectively contributed to the success of stop-word detection. (Figure 7)

As a result, the seamless integration of POS tagging with logistic regression is the optimal strategy for detecting Persian stop words. This method accurately captures the Persian language's complex linguistic nuances and unique grammatical complexities, ultimately leading to precise and effective recognition of stop words. The robust system created by integrating POS tagging with logistic regression effectively addresses linguistic challenges and outperforms other machine learning techniques, providing a solid foundation for Persian stop word recognition. The studies that have employed similar combined methods have shown results consistent with ours, further supporting the effectiveness of integrating POS tagging with machine learning techniques for Persian stop word detection (Raja et al., 2007).

Looking toward the future, integrating POS tagging with logistic regression holds significant promise for advancing stop-word identification in Persian texts. Researchers can develop more reliable, context-aware systems for detecting stop words in Persian texts by combining deep linguistic analysis with advanced machine learning techniques. This future direction mirrors trends in other studies that emphasize the integration of linguistic insights with advanced machine-learning models to improve detection accuracy and efficiency (Dehghani & Manthouri, 2021). This forward-thinking approach aligns with current advancements in natural language processing and opens new avenues for exploring innovative techniques to improve stop-word detection in Persian texts.

In the evolving landscape of language processing, integrating advanced methods with traditional linguistic analysis paves the way for more nuanced approaches to stop word recognition in Persian. By leveraging the strengths of linguistic insights and machine learning, researchers can continue pushing the boundaries of Persian language processing, ultimately leading to more accurate and context-aware systems for stop-word recognition.

Conclusion

The importance of Persian language processing cannot be overstated as the world becomes increasingly connected. Our work provides a solid foundation for future developments in this area. In our search for efficient methods to identify Persian stop words, we conducted a detailed exploration of various approaches and classifiers. Each method brought unique strengths and insights, shedding light on the complex nature of Persian language processing. In summary, our study has caused significant advances in Persian stop-word recognition. It has advanced the field by combining traditional approaches with modern techniques and leveraging an extensive dataset of 30,000 Persian sentences. Similar research using large Persian datasets also highlights the value of combining data-driven techniques with rule-based methods to improve accuracy in Persian stop-word identification (Dehghani & Manthouri, 2021).

These findings enhance our understanding of the Persian language and open new opportunities in areas like text analysis, information retrieval, and sentiment analysis. Looking ahead, researchers in Persian stop word recognition can explore the potential of combining traditional methods with advanced techniques such as deep learning and neural networks to increase the accuracy and efficiency of stop word recognition. Hybrid approaches that leverage the strengths of rule-based methods alongside data-driven models can lead to more robust and context-aware stop-word recognition systems for Persian.

Furthermore, investigating the applicability of pre-trained linguistic models, such as BERT or GPT, tailored to the nuances of the Persian language, could offer new ways to enhance stop word recognition performance. Integrating contextual information and linguistic features with deep learning architectures may help address the challenges posed by Persian stop words' complex and context-dependent nature. This approach aligns with current trends in natural language processing and warrants further exploration to develop effective techniques for stop word detection in Persian texts.

References

Alajmi, A., Saad, E. M. & Darwish, R. (2012). Toward an ARABIC stop-words list generation. *International Journal of Computer Applications,* 46(8), 8-13.

AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M. & Oroumchian, F. (2009). Hamshahri: A standard Persian text collection. *Knowledge-Based Systems*, 22(5), 382-387. https://doi.org/10.1016/j.knosys.2009.05.002

Behera, S. (2018, May). Implement a finite state automaton to recognize and remove stop words in English text on its retrieval. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 476-480). IEEE. https://doi.org/10.1109/ICOEI.2018.8553828

Chanda, S. & Pal, S. (2023). The Effect of Stopword Removal on Information Retrieval for Code-Mixed Data Obtained Via Social Media. *SN Computer Science*, 4(5), 494. https://doi.org/10.1007/s42979-023-01942-7

Chekima, K. & Alfred, R. (2016). An automatic construction of Malay stop words is based on the aggregation method. In *Soft Computing in Data Science: Second International Conference, SCDS 2016, Kuala Lumpur, Malaysia, September 21-22, 2016, Proceedings 2* (pp. 180-189). Springer Singapore.

Chiche, A. & Yitagesu, B. (2022). Part of speech tagging: A systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9, 10. https://doi.org/10.1186/s40537-022-00561-y

Choi, W., Yoo, K. & Choi, S. (2019). Create a List of Stopwords and Typing Errors by TF-IDF Weight Value. *EasyChair*. Retrieved from file:///C:/Users/Reza/Downloads/EasyChair-Preprint-1410.pdf

Daowadung, P. & Chen, Y. H. (2012, July). Stop word in readability assessment of Thai text. In *2012 IEEE 12th International Conference on Advanced Learning Technologies* (pp. 497-499). IEEE. https://doi.org/10.1109/ICALT.2012.9

Dar, K. S., Shafat, A. B. & Hassan, M. U. (2017, June). An efficient stop-word elimination algorithm for the Urdu language. In *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON*) (pp. 911-914). IEEE. https://doi.org/10.1109/ECTICon.2017.8096386

Davarpanah, M. R., Sanji, M. & Aramideh, M. (2009). Farsi lexical analysis and stop word list. *Library Hi Tech*, 27(3), 435-449. https://doi.org/10.1108/07378830910988559

Dehghani, M. & Manthouri, M. (2021, October). Semi-automatic detection of Persian stopwords using FastText library. In 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE) (pp. 267-271). IEEE. https://doi.org/10.1109/ICCKE54056.2021.9721519

Gunasekara, S. V. S. & Haddela, P. S. (2018, October). Context-aware stopwords for Sinhala Text classification. In *2018 National Information Technology Conference (NITC)* (pp. 1-6). IEEE. https://doi.org/10.1109/NITC.2018.8550073

Hao, L. & Hao, L. (2008, December). Automatic identification of stop words in Chinese text classification. In *2008 International conference on computer science and software engineering* (Vol. 1, pp. 718-722). IEEE. https://doi.org/10.1109/CSSE.2008.829

Haque, R. U., Mridha, M., Hamid, M. A., Abdullah-Al-Wadud, M. & Islam, M. S. (2020). Bengali stop word and phrase detection mechanism. *Arabian Journal for Science and Engineering*, 45, 3355-3368. https://doi.org/10.1007/s13369-020-04388-8

Jayashree, R., Murthy, K. S. & Anami, B. S. (2014). Effect of stop word removal on the performance of naïve Bayesian methods for text classification in the Kannada language. *International Journal of Artificial Intelligence and Soft Computing*, 4(2-3), 264-282. https://doi.org/10.1504/IJAISC.2014.062824

Jefriyanto, J., Ainun, N. & Al Ardha, M. A. (2023). Application of naïve bayes classification to analyze performance using stopwords. *Journal of Information System, Technology and Engineering*, 1(2), 49-53. Retrieved from https://pdfs.semanticscholar.org/866d/e716b31fdeb3aa639b3acd9a4902cefce14d.pdf?_gl=1*1127od5*_gcl_au*MTg2NjY3NDM0My4xNzMzNTQ5ODEz*_ga*MTUzNTM3MTM0MS4xNjUyMDMyOTM4*_ga_H7P4ZT52H5*MTczNDA3MDE1NC4xMDAuMS4xNzM0MDc1ODA4LjYwLjAuMA.

Kholwal, R. (2023). Text-Classify: A comprehensive comparative study of logistic regression, random forest, and knn models for enhanced text classification performance. *International Journal of Advances in Engineering & Technology*, 16(5), 415-433. https://doi.org/10.5281/zenodo.10148008

Luhn, H. P. (1985). *Keyword-in-context index for technical literature*. *American Documentation*, 11(4), 288.295. https://doi.org/10.1002/asi.5090110403

Madatov, K., Bekchanov, S. & Vičič, J. (2022). Automatic detection of stop words for texts in the Uzbek language. *Informatica*, 47,143-150. https://doi.org/10.31449/inf.v47i2.3788

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstration*s (pp. 55-60). Retrieved from https://aclanthology.org/P14-5010.pdf

Metïn, S. K. & Karaoğlan, B. (2017). Stop word detection as a binary classification problem. *Anadolu University Journal of Science and Technology A-Applied Sciences and Engineering*, 18(2), 346-359. Retrieved from https://dergipark.org.tr/tr/download/article-file/316284

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. https://doi.org/10.48550/arXiv.1301.3781

Miretie, S. G. & Khedkar, V. (2018). Automatic generation of stopwords in the Amharic text. *International Journal of Computer Applications*, 975, 8887.

Namly, D., Bouzoubaa, K. & Yousfi, A. (2019). A bi-technical analysis for Arabic stop-words detection. *Compusoft*, 8(5), 3126-3134. Retrieved from https://ijact.in/index.php/j/article/view/491

Rahimi, Z. & Homayounpour, M. M. (2023). The impact of preprocessing on word embedding quality: A comparative study. *Language Resources and Evaluation*, 57(1), 257-291. https://doi.org/10.1007/s10579-022-09620-5

Raja, F., Tasharofi, S. & Oroumchian, F. (2007). Statistical POS tagging experiments on Persian text. University of WollongongUniversity of Wollongong. Conference Contribution. Retrieved from https://ro.uow.edu.au/dubaipapers/6

Rajkumar, N., Subashini, T. S., Rajan, K. & Ramalingam, V. (2020). Tamil stopword removal based on term frequency. In *Data Engineering and Communication Technology: Proceedings of 3rd ICDECT-2K19* (pp. 21-30). Springer Singapore. https://doi.org/10.1007/978-981-15-1097-7_3

Raulji, J. K. & Saini, J. R. (2016). Stop-word removal algorithm and its implementation for Sanskrit language. *International Journal of Computer Applications*, 150(2), 15-17. https:doi.org/10.5120/ijca2016911462

Sadeghi, M. & Vegas, J. (2014). Automatic identification of light stop words for Persian information retrieval systems. *Journal of Information Science*, 40(4), 476-487. https://doi.org/10.1177/0165551514530655

Saif, H., Fernandez, M. & Alani, H. (2014, October). Automatic stopword generation using contextual semantics for Twitter sentiment analysis. In *CEUR Workshop Proceedings* (Vol. 1272).

ul Haque, R., Mehera, P., Mridha, M. F. & Hamid, M. A. (2019, May). A complete Bengali stop word detection mechanism. In *2019, the Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and the 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 103-107). IEEE. https://doi.org/10.1109/ICIEV.2019.8858544

Wilbur, W. J. & Sirotkin, K. (1992). The automatic identification of stop words. *Journal of Information Science*, 18(1), 45-55. https://doi.org/10.1177/016555159201800106

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

Zheng, G. & Gaowa, G. (2010, October). The selection of Mongolian stop words. In *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems* (Vol. 2, pp. 71-74). IEEE. https://doi.org/10.1109/ICICISYS.2010.5658841