International Journal of Advanced and Applied Sciences

Journal homepage: http://www.ijaas.in

Informational and Statistical Features-based Model for Removal of Stop Words in Multiple Languages

Rachana Sinha [1]*, Dr. Reena Srivastava [2]

[1] *Research scholar, Department of Computer Applications, Babu Banarasi Das University, Lucknow, India*
[2] *Dean, Department of Computer Applications, Babu Banarasi Das University, Lucknow, India*

ARTICLE INFO

Article history:
Received: 11-03-2025
Received in revised form: 04-04-2025
Accepted: 25-05-2025

Keywords:

stop word, informational features, statistical features, cosine similarity, multiple languages, data-driven

ABSTRACT

Stop word removal is critical in different tasks of Natural Language Processing (NLP) as it reduces corpus length and prepares data for down the line processing. Different languages have distinct predefined stop word list. However, some languages lack such predefined lists, complicating research efforts in those languages. There is no standardized method to identify stop words across all languages, and it is even more challenging to identify domain-specific stop words. This gap motivated our research. Our objective was to study the underlying reasons and develop a method to identify stop words in documents of at least two different languages. We utilized the Reuters News dataset for English and the Turkish News dataset for Turkish text. Our model, calculates the inverse document frequency (IDF), self-information, term frequency (TF), positive point wise mutual information (PPMI), context, co-occurrence, and length for each word, representing each word as a vector of these calculated features and the entire document as a set of word feature vectors. Experimentally, we determined two sets of threshold values for each feature and created two label vectors for stop words and non-stop words. By comparing the word feature vector with these label vectors using cosine similarity, we assigned labels based on the higher similarity value. We validated our model through three different methods and found that it accurately identified all the stop words in the NLTK library for both English and Turkish. Additionally, it was able to recognize domain-specific stop words and other relevant stop words that were missed during the preprocessing phase. These results highlight the potential of our model to be applied to other languages, paving the way for the creation of more comprehensive stop word lists for many under-researched languages..

Introduction

Stop words are the words that are said to carry less important information when it comes to indexing the document or searching the document based on keywords and hence are removed from the document during the pre-processing step. These words are an important part of the structure of a sentence and are found in abundance in a document taking more storage space. There are various types of languages and every language has its own set of stop words. Depending upon the context of the document, some specific words are also taken as stop words, contributing to the variance in the stop word list of even the same language. Stop words need not always be removed. Sometimes removing a stop word that is a part of a product, company

or another name may result in an invalid search of a valid word. Sometimes not removing the stop word also causes an invalid search like not removing a stop word that is plural has a punctuation mark or is misspelled. There is a need for a language-independent universal technique for identifying the stop words in a document. To devise such a universal technique, we need to understand the problems related to words and their morphological forms in the context of different language types.

Morphologically, a word is defined as a unit that can be moved as a whole. Extraneous material cannot be introduced into the middle of a word form. There is a fixed order of morphemes within word forms. Morphological categorization of words relies on how words are constructed and the different ways affixes are added. According to this classification, language can be categorized a

Analytical/ Isolating Language: This type of language is characterized by sentences that are composed of entirely free morphemes. No affixation is allowed. Example: Mandarin Chinese, Vietnamese, Thai, Khmer, Lao

Synthetic language: In this type of language, sentences are built up of different morphemes, and affixation is allowed. It can be further divided into the following categories.

Agglutinative Language: Words in this language are built up of different morphemes that are loosely arranged and can be easily separated. Example: Turkish, Swahili, Punjabi, Swedish, Hungarian, Aztecan, Tamil, Japanese, Korean, Arabic Gujarati, Marathi, Tibetan

Fusional Language: In this language, single words often contain multiple bits of grammatical information due to combining several morphemes. There isn't a straightforward one-to-one relationship between morphemes and the grammatical details they represent. Examples: Russian, Spanish, Greek, Polish, Ukrainian, Ancient French, Sanskrit.

Polysynthetic Language: This type of language is characterized by a high number of morphemes per word. Multiple stems in a single word may be present. Example: Eskimo languages like west Greenlandic, Han

Mixed Language: Language that cannot be divided into the previous four categories is called Mixed language. It shows mixed behavior. Example: English

The variance in the morphological structure of a word and the language it represents makes it difficult to devise a common morphological-based technique for categorizing a word to be a stop word or not. Our hypothesis posits that informational and statistical features of words can be more effective in identifying important words in a document compared to phonological and morphological properties, which are often language-dependent. Notably, these features are language-independent and can be applied to documents in various languages. Our classification model, which identifies stop words and non-stop words, has demonstrated high accuracy rates, achieving 98% for English and 99% for Turkish. These results are significant because English and Turkish belong to different language categories, thus proving the model's suitability across diverse languages. However, our model relies on pre-identified word boundaries of the studied languages.

This paper explores the proposed method, its implementation, and the impact of effective stop word removal on different NLP applications, providing a comprehensive analysis and validation of the approach across multiple languages and datasets. Our model shows promise for creating stop word lists for low-resource languages, thereby contributing to the advancement of NLP in diverse linguistic contexts.

We have discussed the feature considered in detail in Section 3 preceded by Section 2 which talks about the important work done in this area. Section 4 talks about the details of the proposed framework and the algorithm. Section 5 presents the details of the experiment. The results of the study have been discussed in Section 6 followed by the Conclusion and Future Work in Section 7.

Previous work done in the area

In any natural language, the words that appear most repeatedly but do not add much meaning to the text during analysis are called stop words. Some of the domain-specific words are also considered stop words. These words are usually removed from the text before further processing the data. In this section we shall review several papers that discuss innovative methods for removing stop words in various languages, such as Arabic, Chinese, English, French, and Sanskrit.

For Arabic language, researchers developed a stop word removal algorithm using a Finite State Machine. This model achieved 98 percent accuracy. In Chinese [Zou, 06], an algorithm was proposed that was probabilistic, automatically aggregated method-based, though it wasn't tested in other languages. They used a mix of domain-specific and general terms.

For Spanish [Barrón-Cedeno, 09], a new method was introduced to selectively remove stop words from term candidates. Some terms included stop words, which are typically removed using a predefined list. However, researchers suggested only removing the stop word part and keeping the rest if it's a significant candidate phrase. Some of the researchers experimented with different stop word lists. In English and French [Dolamic, 10], a small stop word list of 9 words performed similarly to a larger list of 571 terms, while in Hindi and Persian, a larger list improved results. It indicates clearly that there is no specific rule for creating fixed size list. Then some academics experimented with a hybrid dictionary-based stop word removal algorithm for Sanskrit [Raulji, 16], also achieving 98 percent accuracy. Various stop word removal strategies, including classical methods, Zipf's law [Kaur, 18], mutual information, and term-based random sampling, were systematically reviewed by researchers.

Some of the researchers focused on Indian languages [[Ladani, 20], [Sahu, 22]]. They initially studied the impact of stop words on retrieval performance, then explored various techniques, and developed automatic generation methods for different Indian languages. This early work laid the foundation for advanced natural language processing (NLP) techniques used in sentiment analysis, named entity recognition, and part-of-speech tagging. Academics have discussed significant stop word identification algorithms used by researchers over the past decades for Indian Language and Information Retrieval applications.

Earlier some researchers focused on reducing the word to its root form through stemming that can identify the stop words effectively. In their research [Pandey, 09], some scholars examined the effects of stemming and stop-word removal on Hindi text retrieval. They looked into how stop words affected Hindi text categorization performance and proposed a stop word removal strategy for Hindi text retrieval. For Tamil text classification [Rajkumar, 20], various stop word removal strategies were compared, and a comparative study was done to understand stop word removal techniques in Malayalam [Kumar, 22] and Bengali [Haque, 21].

Researchers have also investigated several stop word identification and removal techniques in several languages. One publication [Sahu, 22] explored the automatic creation of a general stop word list for Hindi text, showing how removing these common terms could improve corpus indexing and information retrieval system performance. In a study [Ashish, 14], researchers created a frequency list from a Gujarati corpus and analyzed well-known Gujarati newspapers, and successfully eliminated stop words from the Gujarati language. Some researchers [Rakholia, 16] used Zipf's Law to generate stop words, while others presented a rule-based method for

dynamically detecting stop words in Gujarati, implemented alongside the Vector Space Model based on cosine similarity for information retrieval in Gujarati [Rakholia, 17].

The scholars [Kaur, 15], [Kaur, 15b],[Kaur, 16]] created the list of Punjabi stop words, including their part-of-speech classifications in Gurumukhi and Shahmukhi scripts. Furthermore, an in-depth analytical study on statistical measures for ceating stop word lists based on continent and script-wise divisions for major foreign languages has been conducted by the researchers in [Saini, 16].

The use of a predefined stop word list to filter stop words from documents, hybrid lists, small and large stop word lists, creating a stop word dictionary, and designing a finite state machine for detecting stop word patterns are some remarkable works in the area of stop word identification and removal. Our investigation, based on the mentioned literature, indicates that each language has its own set of stop words. Even within the same language, the set of stop words can vary depending on the document's context. As a result, researchers have developed novel stop word removal strategies for different languages with positive results. However, there's still a need for language-independent universal techniques for stop word removal.

Informational and Statistical features used in the proposed model

A word is the smallest unit of meaning in a sentence, and it serves an important function in both the syntax and semantics of a sentence in any language. The rules for how words are pronounced are specific to each language and are sensitive to word boundaries. Words are considered to be units of structure and organization, and in most languages, each content word has a primary stress. In some languages, such as Turkish, there is a phenomenon called vowel harmony that applies to the entire word, meaning that vowels can only come before or after the word, but never in the middle.

Additionally, certain sound sequences are not allowed within syllables but may occur before or after a word. For our study, we have defined a word as a token that is separated by a space within a sentence. We considered this definition of word because of its simplicity and its applicability to many languages in general.

Informational and statistical features of words are essential for natural language processing (NLP) tasks such as information retrieval, machine translation, sentiment analysis, and text classification. As these NLP tasks are data dependent besides being language independent and so are the informational and statistical features.These features help in understanding the context and meaning of words, as well as identifying patterns and relationships between them based on

the statistics of the dataset under consideration. Keeping this in mind, we considered the following features.

Term Frequency (TF): A document is a collection of words out of which some words are frequently used for structuring the sentence while some words are rare but gives specific meaning and context to the sentence. Term Frequency refers to the number of times a term (or word) appears in a document. It indicates the importance of a word within a specific document by this count. Very high term frequency shows that the word is less significant in the context of that document and it may be used for structuring the sentence. The formula for TF is:

TF = (Number of times the word appears in the document) / (Total number of words in the document)

Inverse Document Frequency (IDF): In a document the common words and the important words should be distinguished. To weigh down the frequently occurring words and to weigh up the rare but important words in a document, inverse Document Frequency is calculated. This feature measures the rarity of a word across all documents in a corpus. It is calculated as the logarithm of the total number of documents divided by the number of documents containing the word. The formula for IDF is:

IDF = log (Total number of documents in the corpus / Number of documents containing the word)

Context :A word can exhibit different meaning when surrounded by different words. It is important to understand these neighbouring words to understand the usage of the particular word in a sentence. These words that come immediately before and after a particular word are called context of that particular word. We used bi gram technique to calculate the context of a word. In this technique we considered two consecutive neighbouring words as context, then the frequency of these words were counted and the maximum frequency was considered.

Co-occurrence: To understand the contextual relevance and semantic association of a word with another words we calculated Co-occurrence. It measured how often two words appear together within a given context. In our research, we calculated average co-occurrence using a sliding window technique with a window size of five words. This larger window size was chosen to capture broader contextual relationships. Stop words typically exhibit high co-occurrence values.

Positive Pointwise Mutual Information (PPMI): To know the strength of the relationship between two words, we used PPMI that quantifies this strength by comparing their observed co-

occurrence with their expected co-occurrence. It was particularly useful for identifying related terms and improving the accuracy of NLP models. Stop words generally have low PPMI values. The formula for PPMI is:

$$PPMI(x, y) = \max(\log(P(x, y) / P(x) * P(y)), 0)$$

where P(x, y) is the probability of observing x and y together, and P(x) and P(y) are the probabilities of observing x and y separately.

Self-Information: The amount of information a word provides about itself is calculated by Self-information. This feature is often used in information theory to quantify the predictability of a word. With the help of this feature we understood how much a word contributed to the overall meaning of a sentence. Words with high self-information were more informative and carried more weight in the context whereas the stop words had low self-information value. We used the following formula for Self- Information: $I(x) = -\log(P(x))$

Length of a word: The number of characters in a word tells us the length of that word. It can provide information about its complexity or specificity. We found it useful for identifying complex or domain-specific vocabulary in a corpus, which helped us to improve the accuracy of NLP model. Generally the stop words tend to be less lengthy than other important words in a document.

Cosine Similarity: We represented the words as vectors of the above calculated features and considered the threshold values of all the features to create the vectors for the two labels such as SW(stop word) and NSW(non stop word). We wanted to know how the word vector is related to the label vectors. So we used Cosine similarity to determine the similarity between two vectors. It specifically assessed how closely the vectors aligned in terms of their direction, rather than their magnitude. This nearness of a word vector with the specific label vector determined that whether a specific word is a stop word or not. The formula for cosine similarity is $\text{Cosine Similarity} = \cos(\theta) = A.B / ||A|| \times ||B||$

where A.B represents the dot product of vectors A and B, and ||A|| and ||B|| denote the magnitudes (Euclidean norms) of the vectors, and θ is the angle between them.

The above mentioned informational and statistical features of words were crucial for developing accurate and effective NLP model for identifying the stop words in different language text equally. The status of being high or low for the considered feature have been taken based on the general nature of stop words. If it varies for some other language, the status can be adjusted

accordingly without effecting the final results. Since these are dataset dependent characteristics of words, so they can help to improve the performance of various NLP tasks.

The Proposed Framework and the Algorithm

A framework is a set of guidelines or tools that can be used to approach a specific task or problem. Our framework for filtering stop words from text in multiple languages using one model is an example of such a framework. The importance of our framework lies in its ability to automate the process of identifying stop words in text from multiple languages, which can save time and increase efficiency of the model. The framework uses a combination of informational and statistical features to identify stop words, which increases its accuracy compared to traditional rule-based stop word lists. The framework has been pictorially represented in Fig.1 and the explanation follows in the next paragraphs.

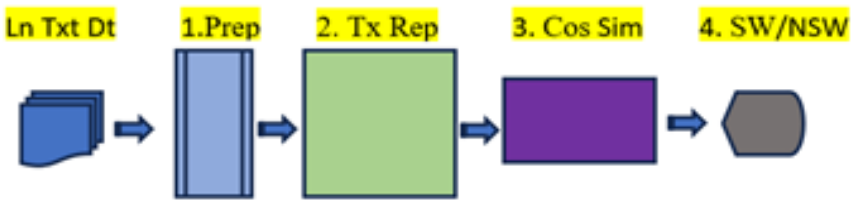


Figure 1: Stages of the proposed framework

Prep: The model starts by preprocessing the text corpus(Ln Txt Dt) by converting it to small case, tokenizing on word level and sentence level, removing numbers, punctuation, and special symbols.

Txt Rep: Next, the model calculates various features for each word, including self-information, term frequency, inverse document frequency, positive point-wise mutual information, co-occurrence, context, and length. These features are used to represent each word as a vector.

Cos Sim: The model then defines two labels, STOP WORD and NON STOPWORD, as vectors of threshold values for each feature to distinguish stop words from non-stop words. Using cosine similarity, the model finds the nearness of the word vector with the label vectors and assigns the respective label to the word based on the maximum value of the cosine similarity.

SW/NSW: This is the last stage of the proposed model where the words labeled as SW are removed from the text documents. The resultant output is the filtered corpus significantly reduced in size and devoid of stop words.

Proposed Model

Data Preprocessing:

Tokenization: Given a document D, represent it as a set of words D = {w_1, w_2, ..., w_n}.

Cleaning: Remove punctuation, special characters, and numbers from each word.

Convert all words to lowercase: w_i = lowercase(w_i).

Feature Extraction:

For each word w_i in document D and corpus C

Term Frequency (TF):TF(w_i) = count(w_i in D) / |D| Inverse Document Frequency (IDF):IDF(w_i) = log(|C| / (|{D in C : w_i in D}| + 1))

Self-Information (SI): SI(w_i) = -log_2(TF(w_i)

Context: Context(w_i) = {w_j : |j - i| ≤ W, j ≠ i} (where W is window size)

Co-occurrence:Cooc(w_i, w_j) = 1 if w_j in Context(w_i), 0 otherwise

Pointwise Mutual Information (PPMI):PPMI(w_i, w_j) = max(0, log_2(P(w_i, w_j) / (P(w_i)P(w_j))))

Length: Length(w_i) = |w_i| (number of characters in w_i)

Feature Vector:

Represent each word w_i as a feature vector: v_i = [TF(w_i), IDF(w_i), SI(w_i), |Context(w_i)|, Σ_j Cooc(w_i, w_j), Σ_j PPMI(w_i, w_j),Length(w_i)]

Label Vector (Median Method):

Calculate the median value for each feature j: median_j = median({v_i[j] : for all w_I having NLTK label1 and NLTK label 0 separately})

Define label vectors: v_stop = [median_0, median_1, ..., median_6] v_non_stop = [median_0, median_1, ..., median_6]

Word Classification (Cosine Similarity)

Calculate cosine similarity between word vector v_i and label vectors: sim_stop = (v_i • v_stop) / (||v_i|| ||v_stop||) sim_non_stop = (v_i • v_non_stop) / (||v_i|| ||v_non_stop||) Classify word w_i If sim_stop > sim_non_stop, label w_i as "STOP". Otherwise, label w_i as "NON-STOP".

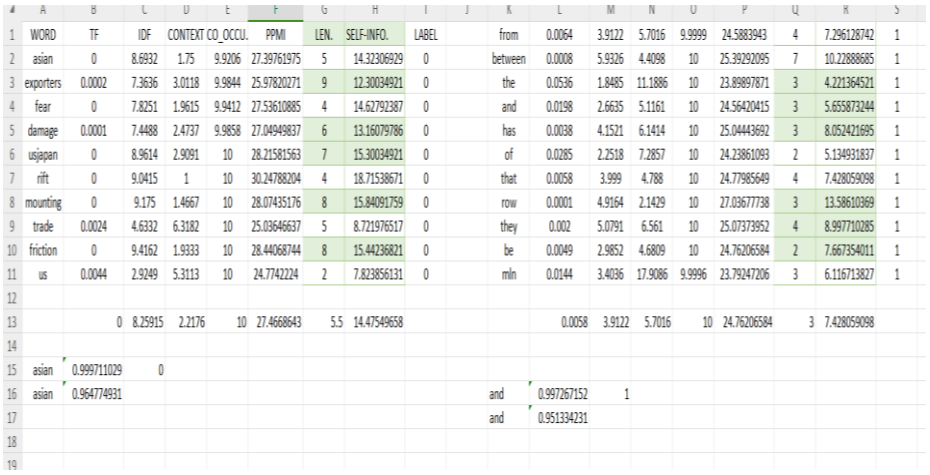
Our model provides an efficient and effective method for filtering stop words from text corpus of multiple languages using a single model. It leverages various features of each word to represent it as a vector and uses cosine similarity to assign the respective label of STOP WORD or NON-STOP WORD. It also proves its reliability by performing equally well through an alternative stage of training the ML model for the desired results. This model can be used in various natural language processing tasks where filtering stop words is necessary, such as text classification, sentiment analysis, and topic modeling.

Experiment and Result Discussion

In this segment, we present the dataset, evaluation criteria, baseline model, and implementation specifics. Subsequently, we perform comprehensive experiments to assess our proposed model and benchmark it against alternative methods. For English language text, we used the Reuters21785 dataset for our experiment. The dataset consists of 21,578 news articles collected from the Reuters financial newswire service in 1987. Articles are categorized into 135 topics, such as "earn" (earnings), "money-fx" (foreign exchange), and "acq" (acquisitions). Initially, we had 12,91,260 words but after preprocessing we were left with 34813 words to train our model. For considering one more language, we took the Turkish News Articles dataset. The dataset includes 3,600 news articles from six Turkish news portals. Articles are categorized into six topics: economy, culture, health, politics, sports, and technology. The dataset is split into training, validation, and test sets, with each article labeled by its category. Each article is labeled with one or more topics, and the dataset is split into training and test sets.

After cleaning the dataset, we calculated all the proposed features for each word. To create the two LABEL VECTORS, we separated the words on the basis of the NLTK labels 0 and 1. Then we took the median value of each section and thus we get the label vectors for SW(stop word) and NSW(non-stop word). We experimented with average values, median values, and standard

deviation values to consider these as threshold values for the features. Among these, the median value taken as threshold gave the best response of about 90% accuracy. This result encouraged us to experiment with machine learning models for training those to get the optimum threshold values. The following figure 2 shows the threshold taken for the experiment with seven features.. In figure 3, B13:M13 shows the label_vector_NSW of median values of all the features for non-stop words, and the range L13:R13 shows the label_vector_SW of median values of all the features for stop words. Cell B15 shows the cosine similarity of the word vector(word: asian) and the label_vector_NSW which is clearly greater than the cosine similarity of the same word vector with the label_vector_SW and thus assigning the label 0 (for non-stop word) to the word. In the same way, the cosine similarity between the word vector(word: and) and the label_vector_SW in the cell L15 is greater than the cosine similarity of the same word vector with the label vector_NSW, hence assigning the label 1 (for stop word) to the word.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	WORD	TF	IDF	CONTEXT	CO_OCCU.	PPMI	LEN.	SELF-INFO.	LABEL		from	0.0064	3.9122	5.7016	9.9999	24.5883943	4	7.296128742	1
2	asian	0	8.6932	1.75	9.9206	27.39761975	5	14.32306929	0		between	0.0008	5.9326	4.4098	10	25.39292095	7	10.22888685	1
3	exporters	0.0002	7.3636	3.0118	9.9844	25.97820271	9	12.30034921	0		the	0.0536	1.8485	11.1886	10	23.89897871	3	4.221364521	1
4	fear	0	7.8251	1.9615	9.9412	27.53610885	4	14.62792387	0		and	0.0198	2.6635	5.1161	10	24.56420415	3	5.655873244	1
5	damage	0.0001	7.4488	2.4737	9.9858	27.04949837	6	13.16079786	0		has	0.0038	4.1521	6.1414	10	25.04443692	3	8.052421695	1
6	usjapan	0	8.9614	2.9091	10	28.21581563	7	15.30034921	0		of	0.0285	2.2518	7.2857	10	24.23861093	2	5.134931837	1
7	rift	0	9.0415	1	10	30.24788204	4	18.71538671	0		that	0.0058	3.999	4.788	10	24.77985649	4	7.428059098	1
8	mounting	0	9.175	1.4667	10	28.07435176	8	15.84091759	0		row	0.0001	4.9164	2.1429	10	27.03677738	3	13.58610369	1
9	trade	0.0024	4.6332	6.3182	10	25.03646637	5	8.721976517	0		they	0.002	5.0791	6.561	10	25.07373952	4	8.997710285	1
10	friction	0	9.4162	1.9333	10	28.44068744	8	15.44236821	0		be	0.0049	2.9852	4.6809	10	24.76206584	2	7.667354011	1
11	us	0.0044	2.9249	5.3113	10	24.7742224	2	7.823856131	0		mln	0.0144	3.4036	17.9086	9.9996	23.79247206	3	6.116713827	1
12																			
13		0	8.25915	2.2176	10	27.4668643	5.5	14.47549658				0.0058	3.9122	5.7016	10	24.76206584	3	7.428059098	
14																			
15	asian	0.999711029	0																
16	asian	0.964774931									and	0.997267152	1						
17											and	0.951334231							
18																			
19																			

Figure 2: Creation of label vectors

To authenticate our hypothesis of creating and using the informational and statistical features for identifying the stop words in a long text, we experimented with these features and trained several machine learning models to classify the words in stop words or non-stop words. To keep the reduced data, we performed feature selection on the calculated features. For this we used SelectKBest, PCA, and Logistic Regression models. We selected the five features Self-information, IDF, Context, PPMI, Length which all the three feature selection models suggested in common.

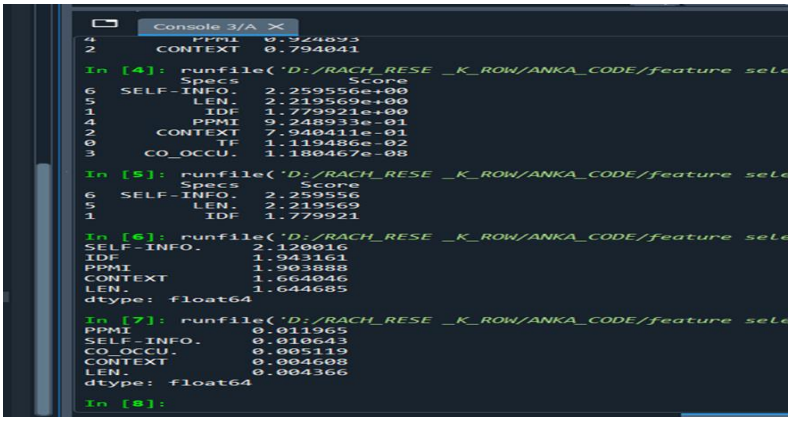


Figure 3: Outcome of Feature Selection Techniques.

After hyper-tuning the models several times we were able to get the accuracy of 96%. Decision Tree achieved an accuracy of 91%, indicating good overall performance. Its precision and recall values are consistent across both classes, suggesting a balanced classification capability. Random Forest showed a slight improvement over the Decision Tree, with 92% accuracy. It also demonstrated balanced precision and recall across classes. K-Nearest Neighbors performed similarly to Random Forest with 92% accuracy. It achieved particularly high precision for Class 0 (97%). Naive Bayes attained a higher accuracy of 94% compared to previous models. It maintained a good balance between precision and recall across both classes. Logistic Regression reached the highest accuracy among individual models at 95%. Gradient Boosting also achieved 95% accuracy but it was not good at classifying non stop words. SVM had a similar performance to Gradient Boosting with 95% accuracy and it was able to identify non stop words as well. MLP (Multi-Layer Perceptron) demonstrated similar performance to Gradient Boosting and SVM, reaching 95% accuracy but its precision and recall for non stop words were low. Ensemble (Previous) method achieved the highest overall performance with 96% accuracy. It displayed excellent precision and recall for both classes, that indicated its robust classification capabilities.

This paragraph discusses about the settings done during the experimentation. The MLP model was designed with four hidden layers. Layer 1 contained 1024 neurons, and used the ReLU activation function. Layer 2 was designed with 512 neurons, and used the ELU activation function. Layer 3 had 256 neurons with the tanh activation function. Layer 4 had 128 neurons with the ReLU activation function. L1 and L2 regularization were applied to all the layers. The Output Layer had a single neuron with a sigmoid activation function. We used ReLU, ELU, and tanh to introduce non-linearity, so that the model can learn complex patterns. We applied Batch Normalization after each hidden layer to normalize the activations, improving training stability and speed. We used Dropouts to randomly drop out neurons during training, preventing

overfitting and enhancing generalization. L1 and L2 regularization were applied to the weights of hidden layers to reduce overfitting by penalizing large weights. We employed Nadam optimizer for efficient weight updates during training. Binary cross-entropy loss was used to measure the difference between predicted and actual labels in classification. To evaluate the model's performance we considered Accuracy, Precision, Recall, and F1 score. The number of trees in the RF and the no. of boosting stages to be built in GB were set to 100. For SVM, we used rbf as the kernel function. The Logistic Regression (LR) was designed using liblinear optimization algorithm. The ensemble model combined the predictions of the base models (NN, RF, GB, SVM, LR) using a soft voting approach. It gave the final prediction based on the weighted average of probabilities predicted by each base model.

To further validate our results, we reorganized the entire dataset by sorting the columns. We arranged IDF, Length, and Self-information in increasing order and Context and PPMI in decreasing order. The top ten percent (3482) words were examined against the NLTK stop word list. It was found that 98 words were common with the NLTK stop word list of 178 words. Then top twenty percent words were examined against the same list and it was found that 116 words were common with the NLTK list of stop words. Similarly considering top thirty percent (11927) words, it was found that 176 words were common with the NLTK stop word list. After removing the duplicate words it was found that 93% words of the NLTK list were present in the dataset while 7% words were not present in the database. Besides several other words which were not mentioned in the list were present in the top thirty percent stop words.

Conclusions

The removal of stop words from the document is an important step in any NLP task as it reduces the size of the document and makes the document more relevant for further processing. The stop word list varies from language to language because of the different morphological features of the language, the context of the document, and the domain under consideration. Hence there is no common method for identifying stop words in a document. We proposed a hypothesis and a framework to address the gap. The proposed framework, addressed the challenges of stop word removal in multiple languages by using informational and statistical features of words rather than relying on language-dependent morphological properties. This approach was effective for English and Turkish (only these two languages were considered for the experiment), achieving high accuracy rates of 96% and 97.5%, respectively. The model demonstrated the potential for creating stop word lists for low-resource languages, thereby advancing NLP applications in diverse linguistic contexts. However, the result has been

achieved under certain experimental conditions (database, samples, experimental setup) and needs to be checked for other databases of different languages. Our paper presents a promising approach to stop word removal that can be applied to various languages and NLP tasks.

Future Work

In future research, we aim to enhance our model by incorporating dynamic thresholding techniques such as percentile-based thresholding, z-score normalization, and clustering-based methods to adjust thresholds based on dataset characteristics adaptively. This will improve the model's robustness and accuracy across diverse datasets. Additionally, we plan to expand the evaluation of our model to include more languages, particularly low-resource and morphologically rich languages, to validate its language-independent capabilities further. A comprehensive comparison with state-of-the-art models, including BERT-based approaches and other deep learning architectures, will also be conducted to benchmark the performance of our model in various NLP tasks. These advancements will strengthen the model's applicability and scalability, paving the way for its use in a wider range of linguistic and domain-specific contexts.

Reference

1. [Al-Shalabi, 04] Al-Shalabi, R., Kanaan, G., Jaam, J. M., Hasnah, A., Hilat, E.: Stop-word removal algorithm for Arabic language, Proceedings. International Conference on Information and Communication Technologies: From Theory to Applications, Damascus, Syria, 19-23 April 2004, IEEE, p. 545.
2. [Zou, 06] Zou, F., Wang, F. L., Deng, X., Han, S., Wang, L. S.: Automatic construction of Chinese stop word list, Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Stevens Point, WI, USA, April 2006, WSEAS, pp. 1010–1015.
3. [Barrón-Cedeno, 09] Barrón-Cedeno, A., Sierra, G., Drouin, P., Ananiadou, S.: An improved automatic term recognition method for Spanish, International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, March 2009, Springer, pp. 125–136.
4. [Dolamic, 10] Dolamic, L., Savoy, J.: When stopword lists make the difference, Journal of the American Society for Information Science and Technology, Vol. 61, No. 1, 2010, pp. 200–203.

5. [Raulji, 16] Raulji, J. K., Saini, J. R.: Stop-word removal algorithm and its implementation for Sanskrit language, International Journal of Computer Applications, Vol. 150, No. 2, 2016, pp. 15–17.
6. [Kaur, 18] Kaur, J., Buttar, P. K.: A systematic review on stopword removal algorithms, International Journal on Future Revolution in Computer Science & Communication Engineering, Vol. 4, No. 4, 2018, pp. 207–210.
7. [Ladani, 20] Ladani, D. J.: Stopword Identification and Removal Techniques on TC and IR applications: A Survey, 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6-7 March 2020, IEEE.
8. [Pandey, 09] Pandey, A. K., Siddiqui, T. J.: Evaluating the Effect of Stemming and Stop-word Removal on Hindi Text Retrieval, First International Conference on Intelligent Human Computer Interaction, Allahabad, India, December 2009, Springer, pp. –, https://doi.org/10.1007/978-81-8489-203-1_31.
9. [Rani, 20] Rani, R., Lobiyal, D. K.: Performance Evaluation of Text-Mining Models with Hindi Stopwords Lists, Journal of King Saud University - Computer and Information Sciences, 2020.
10. [Rajkumar, 20] Rajkumar, N., Subashini, T. S., Rajan, K., Ramalingam, V.: Tamil Stopword Removal Based on Term Frequency, Advances in Intelligent Systems and Computing, Springer, 2020.
11. [Kumar, 22] Kumar, S., Saini, J. R., Bafna, P. B., Wadud, M. A., Islam, M. S.: Identification of Malayalam Stop-Words, Stop-Stems, and Stop-Lemmas Using NLP, Smart Innovation, Systems and Technologies, Springer, 2022.
12. [Haque, 21] Haque, R. U., Mridha, M. F., Hamid, M. A., Wadud, M. A., Islam, M. S.: Comparative Analysis of Bengali Stop Word Detection Using Different Approaches, International Conference on Automation, Control, and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, July 2021, IEEE.
13. [Sahu, 22] Sahu, S. S., Pal, S.: Effect of stopwords in Indian language IR, Sādhanā, Vol. 47, 2022.
14. [Ashish, 14] Ashish, T., Kothari, M., Pinkesh, P.: Pre-Processing Phase of Text Summarization Based on Gujarati Language, International Journal of Innovative Research in Computer Science & Technology (IJIRCST), Vol. 2, Iss. 4, July 2014.
15. [Vijayarani, 15] Vijayarani, S., Ilamathi, J., Nithya: Preprocessing Techniques for Text Mining - An Overview, International Journal of Computer Science & Communication Networks, Vol. 5, No. 1, 2015, pp. 7–16.

16. [Rakholia, 16] Rakholia, R. M., Saini, J. R.: A Rule-based Approach to Identify Stop Words for Gujarati Language, Advances in Intelligent and Soft Computing (AISC), Springer-Verlag, Germany, accepted for publication.
17. [Rakholia, 17] Rakholia, R. M., Saini, J. R.: Information Retrieval for Gujarati Language using Cosine Similarity based Vector Space Model, Advances in Intelligent and Soft Computing (AISC), Springer-Verlag, Germany, accepted for publication.
18. [Kaur, 15] Kaur, J., Saini, J. R.: A Natural Language Processing Approach for Identification of Stop Words in Punjabi Language, International Journal of Data Mining and Emerging Technologies, Vol. 5, No. 2, November 2015, pp. 114–120, DOI: 10.5958/2249-3220.2015.00015.4.
19. [Kaur, 15b] Kaur, J., Saini, J. R.: POS Word Class based Categorization of Gurmukhi Language Stemmed Stop Words, 1st International Conference on Information and Communication Technology for Intelligent Systems (ICTIS-2015), Springer International Publishing, Switzerland, Smart Innovation, Systems, and Technologies (SIST), Vol. 51, 2015, pp. 3–10, DOI: 10.1007/978-3-319-30927-9_1.
20. [Kaur, 16] Kaur, J., Saini, J. R.: Punjabi Stop Words: A Gurmukhi, Shahmukhi, and Roman Scripted Chronicle, National Symposium: ACM Women in Research (ACM-WIR-2016), Indore, India, ACM ICPS, 2016, ISBN: 978-1-4503-4278-0.
21. [Saini, 16] Saini, J. R., Rakholia, R. M.: On Continent and Scriptwise Divisions-based Statistical Measures for Stop-words Lists of International Languages, ICIP-2016: Society of Information Processing's Twelfth International Multi-Conference, International Conference on Data Mining and Warehousing (ICDMW2016), Bangalore, India, Procedia Computer Science, Elsevier, Netherlands.