Multimedia Tools and Applications (2024) 83:50047–50059
https://doi.org/10.1007/s11042-023-17205-9

LiHiSTO: a comprehensive list of Hindi stopwords

Swati Rajwal[1]

Received: 4 August 2022 / Revised: 2 June 2023 / Accepted: 21 September 2023 /
Published online: 4 November 2023

Abstract
A preliminary preprocessing step in text analytics is the removal of words with no semantic meaning, otherwise known as stopwords. English stopwords are very easily accessible and created due to the broad usability of the English language. However, a standard list of Hindi stopwords is still missing. This paper proposes an exhaustive list of generic Hindi stopwords and a Python package for easy distribution and usage. The methodology uses a dual mechanism for creating a list of Hindi stopwords. First, the famous English stopwords are collected and translated into meaningful Hindi words (group 1). Second, unique Hindi stopwords from multiple sources are fetched (group 2). Finally, the respective Hindi stopwords from groups 1 and 2 are combined, which resulted in a significantly large set of 820 Hindi stopwords. Additionally, the list of Hindi stopwords is made openly available for use at the Python Package Index (PyPI) repository as a Python package, which is named *LiHiSTO*. With the help of illustrative implementations, it is shown that LiHiSTO provides abstract and easy access to the list of stopwords for users to perform Hindi text analytics.

Keywords Stopwords · Indian languages · Text analytics · Natural Language Processing · Python package

1 Introduction

Hindi is amongst the top five most spoken languages worldwide. Approximately 600 million native Hindi speakers make it the third most widespread language of 2021 [1]. Natural Language Processing (NLP) is the ability of a system to understand and interpret the natural language such as Hindi, English, Chinese, and any other language. Currently, there are various applications of NLP, including but not limited to language translation, search autocorrect and autocomplete, chatbots, voice assistants, information retrieval, text classification, and many others. But the majority of these NLP services are in the English language [2], and hence the person should know English to converse with such systems. Fortunately, over the past few years, there has been an increasing interest in the languages spoken on the Indian subcontinent, especially Hindi. Given many Hindi speakers worldwide, Hindi has

✉ Swati Rajwal
sr2050@cam.ac.uk

[1] University of Cambridge, Cambridge, UK

become a significant language in the digital domain. Therefore, there is a need to provide NLP-based services to the Hindi-speaking population. The first step towards developing such a sound NLP system is creating a comprehensive list of Hindi stopwords to be used in the pre-processing stage of the Hindi NLP model. This paper aims to build a comprehensive list of Hindi stopwords that will help researchers and computer scientists develop accurate NLP services for the Hindi language. Such highly precise services will then be used by the Hindi-speaking population, making NLP an inclusive space.

Stopwords In text analytics-based NLP applications like information classification and retrieval, removing unnecessary or non-content words from a given corpus is preliminary. Such words carry no semantic meaning of their own and are known as stopwords. Every language has a pre-defined set of commonly used stopwords, also known as generic stopwords [3]. For simplicity, English has stopwords such as often (अक्सर), if (अगर), okay (अच्छा), and many others. Since there is no universal set of stopwords for any language, it can also be custom-created in which the analyst or the team chooses specific words as stopwords for the given purpose. Moreover, there is no need to remove stopwords in some applications, such as machine translation [3].

For any language, there exists a primary number of stopwords. The author believes that removing fundamental or the most generic stopwords in a language like Hindi should not be time-consuming. After removing stopwords, approximately 35–45% of the corpus size is reduced [4]. Therefore, an extensive list of stopwords would ensure that a massive range of uninformative words is removed easily to focus on the essential texts, enhancing system performance [3, 5] and precision [3, 6, 7].

Significance of current study There have been few studies in the past that talk about Hindi stopwords, but none of them provide a comprehensive list of stopwords for a semantically challenging language, Hindi. Due to the morphological richness of the Hindi language, the author believes that the number of Hindi stopwords should be larger than the already existing sets of stopwords [8–12]. The present study introduces *LiHiSTO (pronounced: Lee-Hiss-Toh)*, a comprehensive **Li**st of **Hi**ndi **Sto**pwords. Compared to the existing research studies, the primary contributions of the current work include the following innovative aspects:

- Collection of English translated to Hindi stopwords and the essential Hindi stopwords.
- Development of Python library for easy access to the list of Hindi stopwords by the academic and data analytics community.
- LiHiSTO allows removing a much higher number of stopwords from Hindi texts.
- Currently, this is the only available comprehensive study with 820 Hindi stopwords to the best of the author's knowledge.

The development of LiHiSTO is necessary for any Hindi text-based NLP application for removing maximum Hindi stopwords and improving analysis time by focusing on more critical textual data.

Organization of this paper The remainder of the paper is organized as follows: Section 2 presents the related studies in the literature. Section 3 provides the proposed methodology for collecting stopwords from various sources. This section also talks about the Python

processing done on the stopwords. Section 4 discusses the final list of Hindi stopwords and the subsequent development of the Python library for open-source usage. This section also showcases an illustrative implementation and comparative analysis of LiHiSTO. Finally, Section 5 concludes the paper.

2 Literature review

In this section, the author discusses some previously published research studies related to English and non-English stopwords. The primary focus is on the Hindi language, which is the central theme of the current study.

Stopwords are non-informative words such as conjunctions, adverbs, prepositions, articles, etc. Stopwords removal is a useful process that has been in the attention of research scholars for various NLP activities such as text classification and information retrieval [2, 3, 6, 13], text-processing [4, 14], text-mining [5, 8], text-encoding [15], text-categorization [16, 17], and others. The concept of stopwords in the aforementioned areas of NLP areas has been around for more than sixty-five years, focusing on the English language in the much earlier days [15, 18]. In 1957, H.P. Luhn [15] published a study that talked about stopwords and their level of subject specificity. Silva et al. [16] experimentally proved the worthiness of stopwords removal. Their results concluded that removing stopwords is the most influential task in text categorization, and this activity has importance in improving categorization performance. Most of the past and contemporary research works have targeted English stopwords [3, 6]. But recently, there has been an increasing interest of researchers in developing a list of stopwords for various non-English languages such as Arabic, Chinese, Yoruba, Amharic, and Sinhala [3]. A similar trend is observed for Indian languages like Gujarati [6], Bengali [14], and many other non-Indian languages such as Turkish [19]. One such work in developing a stopwords list for Sanskrit was carried out by [13] using an automated algorithm aided with manual intervention by subject experts. Now, Hindi being the most widely spoken non-English Indian language, is no exception where research enthusiasts have dedicated a lot of time to creating generic and domain-specific [5, 8] Hindi stopwords. Studies similar to [16] were carried out by Pandey et al. [2] and Singh et al. [7] for the Hindi language. In [2], the experimental results suggested a significant increase in retrieval performance due to the removal of Hindi stopwords. Similarly, the empirical investigation by [7] demonstrated that removing Hindi stopwords increased word sense disambiguation precision by 54.81%. In the light of various research studies, it is established that a comprehensive list of stopwords will have a remarkable impact on the accuracy and precision of a Hindi NLP-based application such as text processing, mining, classification, categorization, and retrieval.

All the studies, as mentioned above, are an effort to complement the contributions made by researchers worldwide working towards the inclusion of non-English languages (especially Hindi) in NLP research. Although the studies have been influential in promoting more research in Hindi stopwords, there is still a need for a comprehensive list of Hindi stopwords. Moreover, there is also a lack of proper distribution and reproducibility of the work contributed by these studies. Hence, the present work addresses the above-mentioned gaps by rigorously identifying generic, insignificant, uninformative Hindi stopwords and developing a Python-based package for easy usage and free distribution. The current study contributes to the existing body of knowledge and will aid the researchers in quickly preprocessing Hindi texts for building NLP models.

3 Methodology

This section covers the methodology for creating a comprehensive list of Hindi stopwords. First, the lists of English stopwords are collected from various sources, and the processing is done using the powers of the Python programming language. Afterward, the English stopwords are translated into Hindi. Second, the lists of Hindi stopwords are collected from multiple sources and processed via Python for removing duplicates. By the end of this section, two groups of Hindi stopwords are formed, then combined into one, as will be discussed in subsequent sections.

3.1 English-to-Hindi stopwords (Group 1)

As already mentioned, there are various lists of English stopwords available. It is primarily attributed to English being the most spoken language worldwide [1].

3.1.1 Sources of English stopwords

Table 1 lists various sources for English stopwords, their URL, brief description, and the number of English stopwords offered. The availability of mature and almost complete sets of generic English stopwords is the primary reason for selecting the English language in this study. Each source mentioned in Table 1 has an individual text file with the corresponding number of stopwords. All source text files are read into Python's set() variable, which removes any redundancy. This resulted in a list of 824 unique English stopwords. These stopwords are stored in a Google Sheet (GSheet) for archive and automatic translation to Hindi, discussed in the following subsection.

3.1.2 Translation of English stopwords to Hindi

The Unique **824** English stopwords are stored in a GSheet and translated to Hindi using the Google Translate formula on GSheet as shown in Eq. (1):

$$= \text{if (ISBLANK (A1), A1, GOOGLETRANSLATE (A1, } ''\text{en}'', ''\text{hi}''\text{))} \quad (1)$$

The function GOOGLETRANSLATE() takes in the following three parameters:

- **Text**: It is text to translate from source_language to target_language.
- **Source language**: It is a two-lettered language code of the source language. For instance, in Eq. (1), "en" stands for English.
- **Target language**: It is a two-lettered language code of the target language. For instance, in Eq. (1), "hi" for Hindi.

Equation (1) converts the given text (i.e., English stopword in a single GSheet cell) from a source language (i.e., English) to a target language (i.e., Hindi). The list of English stopwords is stored in column A of the GSheet. The formula in Eq. (1) is applied to all cells of column B. As Python code triggers, each of the 824 unique English stopwords is stored in column A of GSheet. Alongside, the GOOGLETRANSLATE() function automatically translates each English stopword to Hindi and simultaneously accumulates it in the corresponding cell of column B. A small set of the GSheet is shown in Table 2.

Table 1 Collection of English Stopwords (SW) from various sources

Source	URL	Description and Selection Criteria	# of English Stopwords
stopwords 1.0.0 [20]	https://pypi.org/project/stopwords/	Popular PyPi library	173
NLTK Package [21]	https://pypi.org/project/nltk/	Popular python package	179
Kaggle [22]	https://www.kaggle.com/rowhitswami/stopwords	Repository with English stopwords	733
Ranks.nl [9]	https://www.ranks.nl/stopwords	Website with stopwords in various languages	173
Total			1258
Total (without duplicates)			824

Table 2 English to Hindi translation by GOOGLETRANSLATE() function results in identical Hindi translation. Hence redundancy is created

English stopword (Column A)	Hindi translation (Column B)
shouldn	नहीं करना चाहिए
shouldn't	नहीं करना चाहिए
shouldnt	नहीं करना चाहिए
serious	गंभीर
seriouser	गंभीर

3.1.3 Human evaluation

It was observed that the GSheet formula (Eq. 1) could not generate an English-to-Hindi translation for multiple entries due to misspelled English words. Therefore, a human evaluation was employed to identify and remove erroneous entries manually. The person to remove the inaccurate entries knows both Hindi and English. Furthermore, some Hindi translations were incorrect, and hence, manual corrections for a few entries were also performed. For instance, the GOOGLETRANSLATE() function of GSheet translated the English word 'till' to 'हमहूं कक्काजी हो जाएंगे ट्वैन्टी फर्स्ट सैन्चुरी तक' in Hindi. This translation is incorrect, and therefore it was manually corrected with 'जब तक'. The manual intervention removed misspelled words and rectified Hindi translations of 109 English stopwords. These stopwords previously resulted in either wrong or no translation in Hindi.

3.1.4 Removal of redundancy

Now, there are 715 English-Hindi stopwords left. But the list of **715** translated Hindi stopwords contains duplicates. Let us take an example to understand why the redundant copies exist in the Hindi translation even though the redundant elements in English stopwords were removed. Table 2 lists a few English stopwords and the corresponding Hindi translation according to the GOOGLETRANSLATE() function of GSheet.

As shown in Table 2, even though the English words are distinct, their Hindi translation means the same. Python provides a straightforward way of removing duplicates in a list of elements. Therefore, the Hindi-translated stopwords stored in GSheet are read into a Python **set()** variable, eliminating duplicate elements. After removing duplicates, a total of **544** unique Hindi stopwords from English stopwords are gathered, collectively referred to as Group 1 in this paper.

3.2 Hindi stopwords (Group 2)

The popular Hindi stopwords are collected from various sources in this group, as listed in Table 3. The table also mentions the source's URL, its description, and the number of stopwords they offer. It can be observed that the number of Hindi stopwords gathered is less than the number of English stopwords (Table 1).

Table 3 mentions the number of Hindi stopwords from each source. There is an individual text file with stopwords from each source containing a list of Hindi stopwords. The contents of each text file are read into Python code for further processing, such as the removal of duplicates using the set() function in Python. This activity resulted in 382 unique Hindi stopwords, named Group 2, from all the sources.

Table 3 Collection of Hindi Stopwords (SW) from various sources

Source	URL	Description	# of Hindi Stopwords
GitHub [10]	https://github.com/stopwords-iso/stopwords-hi	Repository with Hindi stopwords and highest GitHub stars	225
Kaggle [11]	https://www.kaggle.com/kmldas/indian-language-stopwords	Repository with the highest number of Hindi stopwords	264
Kaggle [12]	https://www.kaggle.com/heeraldedhia/stop-words-in-28-languages	Repository with stopwords from 28 languages	169
Ranks.nl [9]	https://www.ranks.nl/stopwords	Website with stopwords in various languages	97
Hindi Alphabets	None	Collection of all Hindi alphabets	49
Total Stopwords			804
Total Stopwords (without duplicates)			382

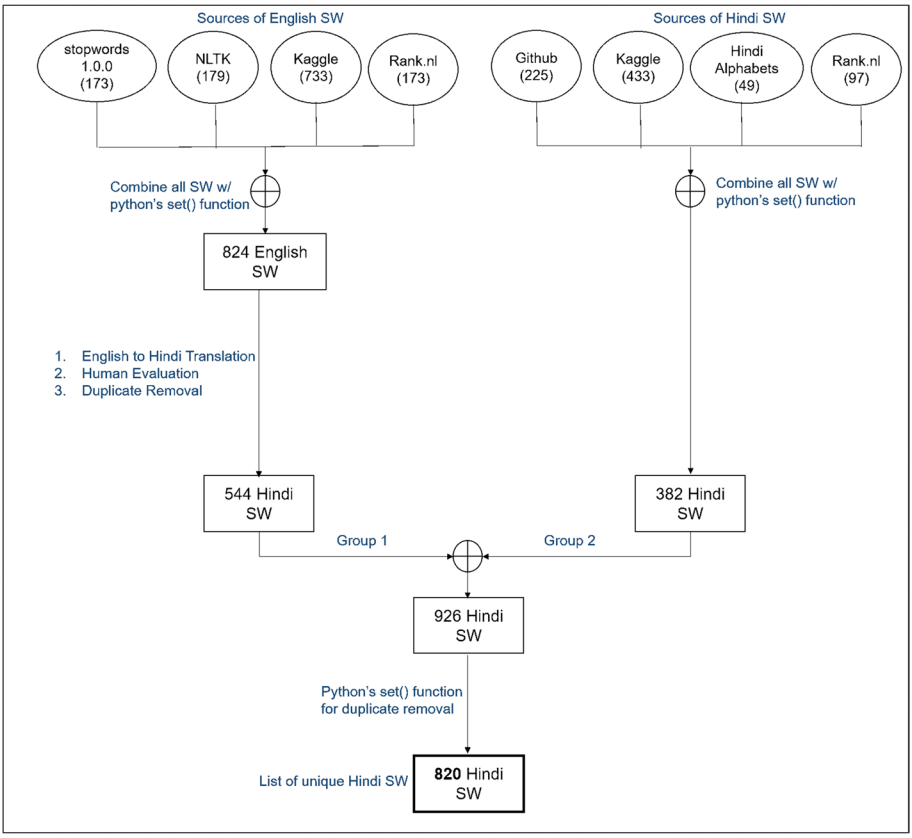

Fig. 1 Collection and processing of Hindi stopwords. SW refers to stopwords in the figure (Also the term 'Kaggle' collectively refers to two lists from the Kaggle platform as mentioned in Table 3). After each source, the numbers in the bracket represent the total number of stopwords derived from the start

4 Results

This section discusses the list of unique Hindi stopwords after combining groups 1 and 2. A pictorial representation of the entire methodology is shown in Fig. 1. The figure shows the collection of English and Hindi stopwords from various sources. English stopwords are converted to Hindi, rectified, and removed any duplicates. Then the Hindi stopwords from both groups are combined into one. Finally, to aid the access to Hindi stopwords, a Python package is developed, discussed in subsequent sections.

4.1 Hindi stopwords list comparison

The list of unique Hindi stopwords in groups 1 and 2, respectively, is combined into one. As expected, there was redundancy, and it was taken care of by the set() function in Python. The final list of unique 820 Hindi stopwords is named LiHiSTO. The number of Hindi stopwords in the present study is compared with other sources, as shown in Table 4.

Table 4 Comparison with various lists of Hindi stopwords

Source	Github	Kaggle1	Kaggle2	Rank.nl	LiHiSTO
# of Hindi stopword	225	264	169	97	820

Also, note that the proposed list of Hindi stopwords has 820 words and phrases, where 1 phrase is of the size of five words, 16 phrases are of the size of four words, 41 phrases are of size three words, 173 phrases are of the size of two words and 589 words are of the size of one word.

4.2 LiHiSTO, the Python package

Python is amongst the top five most widely used programming languages around the world[1]. Python provides easy-to-use data structures and libraries. Moreover, there is a tremendous abundance of documentation and community support for Python. Due to its simplicity and libraries, python is preferred for text analytics. Developers around the world distribute their Python packages on platforms such as Python Package Index (PyPI) [23]. PyPI is a central repository for Python packages and libraries. It allows one to publish and distribute open-source projects that are not part of standard Python libraries. One can find and install software developed and shared by the Python community on PyPI. At the time of this study, there were almost 458,232 projects and 705,945 users associated with PyPI. Pip [24] is the most commonly used package management tool to download and install Python libraries from PyPI.

There is a need for an abstract way to access the list of Hindi stopwords proposed by this study. This would allow the researchers to quickly access the stopwords without any hustle. Since Python is a popular programming language for text analytics, developing a Python package to provide easy access to Hindi stopwords is well-suited. The user simply installs the Python package into their Python code and calls the function which returns the Hindi stopwords list. The installation and accessing the package are discussed in the next section.

4.3 Illustrative implementation

LiHiSTO is developed in Python programming language and is currently hosted at https://pypi.org/project/LiHiSTO/. The package is published under the OSI Approved::MIT License to allow for continuous use and application in text-based analytics. The intended manner to use the library consists of three main steps namely- Installation, Import, and stopwords loading. In order to install the package using pip, run the following Python command:

```
pip install LiHiSTO  # required only one time
```

[1] https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/.

```
from stopwords_hindi import hindi_sw # Import statement
Stopwords = hindi_sw.get_hindi_sw()  # loading stopwords
```

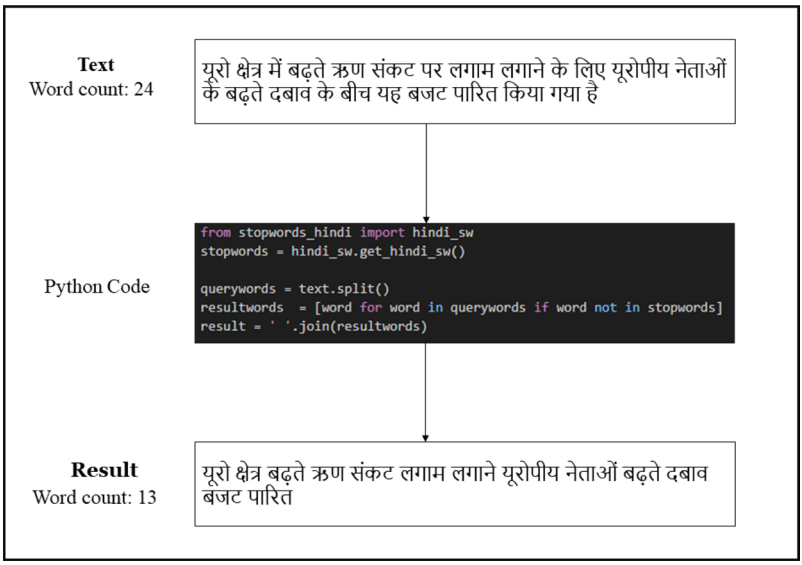

Fig. 2 Removal of stopwords from a Hindi text using LiHiSTO

Once installation is successful, an import statement is required to make the code in LiHiSTO's module available in yours. Thereafter, the function call is made to load a list of Hindi stopwords. The function namely 'get_hindi_sw()' returns a list of Hindi stopwords that can be stored in a Python list. The Python code to import and load the Hindi stopwords to a variable named 'stopwords' is shown below:

```
from stopwords_hindi import hindi_sw # Import statement
Stopwords = hindi_sw.get_hindi_sw()  # loading stopwords
```

Let's take an example to showcase the application of LiHiSTO. For simplicity, a single textual sentence in Hindi is considered here. Once the installation, import, and loading of packages are done, we can start removing stopwords from Hindi text data as shown in Fig. 2. Also, note the reduction of words in the original textual data. The original text contained 24 words. After the removal of stopwords using LiHiSTO, the concerned text was reduced to 13 words only. It means the text has almost 45% of stopwords which is low-level information. Hence, the Hindi text analysis space is now reduced to 55% of the original text. LiHiSTO as a platform can accelerate the Hindi text-based NLP research studies by removing a large chunk of stopwords from the concerned text.

Table 5 Classification results before and after removal of LiHiSTO stopwords

		PR	BH	HDA	BBC
Logistic Regression	Before	0.67	0.60	0.73	0.54
	After	0.64	0.60	0.69	0.55
SVM	Before	0.68	0.61	0.73	0.59
	After	0.64	0.61	0.69	0.59
SGD	Before	0.64	0.59	0.70	0.58
	After	0.61	0.58	0.68	0.58
Decision Tree	Before	0.57	0.58	0.54	0.43
	After	0.53	0.59	0.54	0.43
Random Forest	Before	0.58	0.59	0.63	0.39
	After	0.49	0.58	0.58	0.41

4.4 Comparative analysis

The four datasets used here for showcasing the effects of stopwords on classification accuracy are Product Review (PR) [25], Bhaav (BH) [26], Hindi Discourse Modes Dataset (HDA) [27], and Hindi BBC News Dataset (BBC) [28]. More details about each of the datasets have been specified in this paper [29] where the authors have re-casted the dataset for natural language inference in Hindi.

The textual datasets considered here underwent no preprocessing, except for the elimination of stopwords using the LiHiSTO package. This was done specifically to examine the impact, or lack thereof, of stopwords on accuracy. Stopwords, being essentially ineffectual words, were removed to save space and processing time. Five classification algorithms were employed on the datasets to demonstrate the effects of stopwords. Each dataset was processed twice: once using the original dataset (without removing any stopwords) and then after the removal of stopwords. Following the removal of stopwords, the average reduction in corpus sizes was 50.22%, 44.39%, 33.04%, and 41.44% for the BH, PR, BBC, and HDA datasets, respectively. Table 5 presents the classification accuracy achieved by each algorithm before and after the removal of stopwords. A visual representation of Table 5 is depicted in Fig. 3. It illustrates that the amount of Hindi textual data that the algorithm needs to process is reduced after removing stopwords, while the classification accuracy remains almost unchanged. It is important to mention here that no other preprocessing techniques such as stemming or lemmatization were applied, which could have potentially resulted in higher accuracies. The similar performance across different algorithms signifies the correct identification of Hindi stopwords by the LiHiSTO package.

5 Conclusion

This study proposes a comprehensive list of Hindi stopwords. The designed methodology uses a dual means to collect Hindi stopwords from English-to-Hindi translation (group 1) and actual Hindi stopwords (group 2). After processing group 1, people with knowledge of Hindi and English manually performed two tasks: removing misspelled stopwords and correcting the wrong Hindi translations. After combining the unique stopwords from both groups, a resultant list of 820 unique stopwords in Hindi was created. Furthermore, a Python package called LiHiSTO has been developed and introduced in this study. LiHiSTO provides abstract

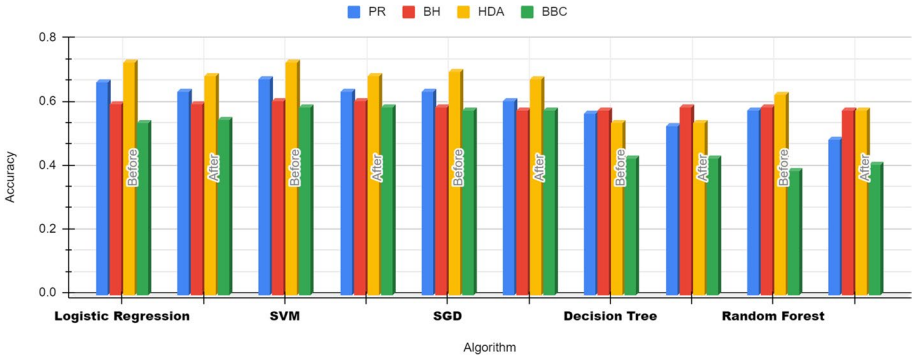

Fig. 3 Visual representation of classification accuracy on four datasets by five classification algorithms. The presence or absence of stopwords has minimal effect on the overall accuracy achieved in each case

and easy access to the list of stopwords for users to perform Hindi text analytics. The application of the package has been showcased through illustrative implementations.

Acknowledgements The author would like to thank the journal reviewers for their valuable suggestions contributing to the improvement of this paper.

Funding None. No funding to declare.

Data availability This paper introduces LiHiSTO, a Python package with a comprehensive list of 820 Hindi Stopwords. The code and other details can be found at https://pypi.org/project/LiHiSTO/.

Declarations

Competing interests The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Published by M. Szmigiera and M 30. Mostspokenlanguagesintheworld. Statista, 30-Mar-2021. [Online]. Available: https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/. Accessed 2 June 2023
2. Pandey AK, Siddiqui TJ (2009) Evaluatingeffectofstemmingandstop-wordremovalonHinditextretrieval. In: Proceedings of the First International Conference on Intelligent Human Computer Interaction. Springer India, New Delhi, pp 316–326
3. Ladani DJ, Desai NP (2020) Stopword Identification and Removal Techniques on TC and IR applications: A Survey. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp 466–472. https://doi.org/10.1109/ICACCS48705.2020.9074166
4. Jha V, Manjunath N, Shenoy PD, Venugopal KR (2016) HSRA: Hindi stopword removal algorithm. 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), pp 1–5. https://doi.org/10.1109/MicroCom.2016.7522593

5. Rani R, Lobiyal DK. Performance evaluation of text-mining models with Hindi stopwords lists. J King Saud Univ-Comput Inf Sci Elsevier BV, Mar-2020 [Online]. Available: https://doi.org/10.1016/j.jksuci.2020.03.003
6. Joshi H, Pareek J, Patel R, Chauhan K (2012) Tostopornottostop—Experiments on stopword elimination for information retrieval of Gujarati text documents. In: 2012 Nirma University International Conference on Engineering (NUiCONE)
7. Singh S, Siddiqui TJ (2012) Evaluating effect of context window size, stemming and stopword removal on Hindi word sense disambiguation. In: 2012 International Conference on Information Retrieval and Knowledge Management
8. Rani R, Lobiyal DK (2018) Social choice theory based domain specific Hindi stop words list construction and its application in text mining. inIntelligent Human Computer Interaction. Springer International Publishing, Cham, pp 123–135
9. Stopwords [Online]. Available: https://www.ranks.nl/stopwords. Accessed 2 June 2023
10. Stopwords-Iso. Stopwords-ISO/stopwords-hi: Hindi stopwords collection. GitHub.[Online]. Available: https://github.com/stopwords-iso/stopwords-hi. Accessed 2 June 2023
11. Das K. Indian language Stopwords. Kaggle, 05-Oct-2020.[Online]. Available: https://www.kaggle.com/kmldas/indian-language-stopwords?select=Hindi_stopwords.txt. Accessed 2 June 2023
12. Dedhia H. Stopwords in 28 languages. Kaggle, 30-Sep-2020.[Online]. Available: https://www.kaggle.com/heeraldedhia/stop-words-in-28-languages. Accessed 2 June 2023
13. Raulji JK, Saini JR (2017) Generating stopword list for Sanskrit language. In: 2017 IEEE 7th International Advance Computing Conference (IACC)
14. Haque RU, Mridha MF, Hamid MA, Abdullah-Al-Wadud M, Islam MS (2020) Bengali stop word and phrase detection mechanism. Arab J Sci Eng 45(4):3355–3368
15. Luhn HP. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. IBM J Res Dev 1(4):309–317, IBM. Oct-1957 [Online]. Available: https://doi.org/10.1147/rd.14.0309
16. Silva C, Ribeiro B (2004) The importance of stopword removal on recall values in text categorization. In: Proceedings of the International Joint Conference on Neural Networks, 2003
17. Coban O (2022) An item response theory-based approach for text categorization. Arab J Sci Eng 47:9423–9439
18. Burchfield R (1985) Frequency analysis of english usage: Lexicon and Grammar. By W. Nelson Francis and Henry Kučera with the assistance of Andrew W. Mackie. Boston: Houghton Mifflin. 1982. x +561. J Eng Linguist 18(1):64–70
19. Bozkurt F, Çoban Ö, BaturalpGünay F, YücelAltay Ş (2019) High performance twitter sentiment analysis using CUDA based distance kernel on GPUs. Tehnički Vjesn 26(5):1218–1227
20. Stopwords PPI [Online]. Available: https://pypi.org/project/stopwords/. Accessed 2 June 2023
21. Bird S, Klein E, Loper E (2009) Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.
22. Ragnar. All English stopwords (700+). Kaggle, 28-Nov-2020. [Online]. Available: https://www.kaggle.com/rowhitswami/stopwords. Accessed 2 June 2023
23. Find, install and publish python packages with the python package index. PyPI. [Online]. Available: https://pypi.org/. Accessed 2 June 2023
24. Pip PPI [Online]. Available: https://pypi.org/project/pip/. Accessed 2 Jan 2022
25. Akhtar MS, Kumar A, Ekbal A, Bhattacharyya P (2016). A hybrid deep learning architecture for sentiment analysis. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 482–493)
26. Kumar Y, Mahata D, Aggarwal S, Chugh A, Maheshwari R, Shah RR (2019) BHAAV- A text Corpus for emotion analysis from Hindi stories. ArXiv. https://doi.org/10.5281/zenodo.3457467
27. Dhanwal S, Dutta H, Nankani H, Shrivastava N, Kumar Y, Li JJ, Mahata D, Gosangi R, Zhang H, Shah RR, Stent A (2020) An annotated data set of discourse modes in Hindi stories. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp 1191–1196, Marseille, France. European Language Resources Association
28. NirantK. (n.d.) Release dataset release: BBC Hindi v0.1 · NIRANTK/hindi2vec. GitHub. Retrieved May 7, 2023 from https://github.com/NirantK/hindi2vec/releases/tag/bbc-hindi-v0.1
29. Uppal S, Gupta V, Swaminathan A, Zhang H, Mahata D, Gosangi R, … Stent A (2020) Two-step classification using recasted data for low resource settings. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pp 706–719