A study on language-independent stemmer in the Indian language IR

Siba Sankar Sahu[1] and **Sukomal Pal**[2]

[1] Department of Computer Science and Engineering,
Sardar Vallabhbhai National Institute of Technology, Surat,
Gujarat, 395007, India
sibasankar@coed.svnit.ac.in
[2] Department of Computer Science and Engineering,
Indian Institute of Technology (BHU), Varanasi,
Uttar Pradesh, 221005, India
spal.cse@itbhu.ac.in

Abstract

We explore and evaluate the effect of different language-independent stemmers in information retrieval (IR) tasks with Indian languages such as Hindi, Gujarati, and English. The issue was examined from two points of view. Does language-independent stemmer improve retrieval effectiveness in Indian languages IR? Which language-independent stemmer is the most suitable for different Indian languages? It is observed that stemming enhances retrieval efficiency in different Indian languages compared to the no stemming approaches. Among the different stemmers experimented with, the co-occurrence-based stemmer (SNS) performs the best and improves a mean average precision (MAP) score by 2.98% in Hindi and 20.78% in Gujarati languages, respectively, while the graph-based stemmer (GRAS) performs the best and improves a MAP score by 5.83% in English.

1 Introduction

In a morphologically rich language, many words go through different kinds of morphological inflections. Different stemmers have been proposed and evaluated to deal with morphological inflections. Stemming is a mechanism that transforms morphological variants of a word to their root form by stripping suffixes and prefixes from the inflected word. For example, 'education', 'educating', 'educated', and 'educational' map to their root word 'educate'. In IR systems, stemming has two benefits. One, stemming reduces index size significantly by conflating several terms. Two, it improves recall by retrieving a large number of potentially relevant documents Hull (1996). In the current state-of-the-art, stemmers are broadly categorized into two types. One is a rule-based stemmer (language-dependent), and another is a statistical stemmer (language-independent). In a language with good linguistic resources, such as English, many rule-based stemming algorithms (Porter (1980) and Lovins (1968)) have been proposed and evaluated. The drawback of rule-based stemmers is that they are language-specific, i.e. a particular stemmer could not be used in other languages. However, statistical stemmers are more beneficial in a language with unknown grammar rules and less availability of linguistic resources. The main benefit of a language-independent stemmer is that it does not require any linguistic knowledge to implement it.

Different rule-based stemmers are proposed and evaluated in different evaluation forums (e.g., CLEF[1], TREC[2], NTCIR[3] and FIRE[4] evaluation campaign) for European, Asian and South Asian languages. In addition to these evaluation campaigns, different stemmers are proposed and evaluated in European and Asian languages. Savoy (2006) proposed different rule-based light stemmers for European languages (French, Portuguese, German, and Hungarian). They observed that stemming improves retrieval performance in the IR domain. Dolamic and Savoy (2010) proposed a light (inflectional) and aggressive (derivational) stemmer in Bengali, Hindi and Marathi. They removed inflectional and derivational suffixes from nouns and adjectives that frequently occurred. They observed that the stemmer improves retrieval performance in the IR domain.

In recent years, different language independent stemmers (Goldsmith (2001), Xu and Croft (1998), Paik et al. (2011a), Paik et al. (2011b)) have been proposed and evaluated for European and few Asian languages. (Majumder et al. (2007), Paik et al. (2011b) and Paik et al. (2011a)) investigated the effect of language-independent stemming techniques in Indian (Bengali and Marathi) and Euro-

[1] http://www.clef-initiative.eu/
[2] https://trec.nist.gov/
[3] http://research.nii.ac.jp/ntcir/
[4] http://fire.irsi.res.in/

Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025, pages 181–189, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-020

pean (Hungarian, French, Czech and Bulgarian) languages IR. They observed that the performance of language-independent stemmers was comparable to that of rule-based stemmers in different Indian and European languages. This study explores and evaluates the effect of language-independent stemmers in Indian languages IR.

We primarily explore the following research questions (RQs).

RQ1: Does language-independent stemmer improve retrieval performance in different Indian languages IR? If yes, to what extent?

RQ2: Which language-independent stemmer is the most suitable for different Indian languages? Whether to use Yet another suffix stripper (YASS) or fast-corpus-based (FCB) or co-occurrence-based (SNS) or graph-based (GRAS), or Trunc-n-based indexing?

Hence, we evaluated different language-independent stemming strategies in Indian languages IR. Moreover, we suggest the best stemming technique in the IR domain. The contributions of this article can be summarised as follows.

1. We investigated different language-independent stemming strategies such as FCB, SNS, GRAS, YASS, or Trunc-n-based indexing in Indian languages IR.

2. The effectiveness of different language-independent stemming strategies is evaluated and compared with no stemming approaches in the IR domain.

3. Analysis has been done for different language-independent stemming strategies and IR models and suggests the best stemming strategy and IR model for different Indian languages.

The rest of the article is organized as follows. Section 2 reviews the state-of-the-art techniques related to stemming methods in the text analysis domain. Section 3 describes the algorithm for implementing language-independent stemmers in Indian languages. Different retrieval models are used in the experimentation is described in section 4. The statistic of the test collection is presented in Section 5. Evaluation result and their analysis is presented in section 6. Finally, we conclude with directions for future work in section 7.

2 Related Work

Porter (1980) and Lovins (1968) are the two most popular rule-based stemmers built in English. In the Porter stemmer, the suffixes are truncated sequentially. Similarly, the Lovin stemmer removes suffixes by implementing 35 rules. They looked at 294 suffixes, and the longest suffix was eliminated at first. The Dawson (1974) stemmer worked like the Lovin stemmer, but comprised a larger number of suffixes, that is, 1200. The Dawson stemmer is built to remove errors in the Lovin stemmer. These rule-based stemmers improve retrieval performance in the IR domain. Hull (1996) observed that the stemmer performs moderately in English and does not produce statistically significant results. Many rule-based stemmers have been proposed and evaluated in different low-resource languages. We outline a few stemming techniques in the following.

Recently, there has been a substantial growth of Non-English languages on the Web. These Non-English languages require an efficient preprocessing technique to improve the performance of an IR system. Hence, the researchers organized different evaluation campaigns, proposed different stemming techniques, and evaluated their effectiveness in the IR domain. In the CLEF evaluation campaign, Peters (2008) proposed different stemmers for European languages. Similarly, in the NTCIR evaluation campaign, different stemmers are presented and evaluated in Japanese, Korean, and Chinese languages. FIRE [5] organized different shared tasks and proposed different stemming techniques for South Asian languages. The performance of these stemmers is evaluated in the monolingual and cross-lingual retrieval domain. Sahu et al. (2023) evaluated the effect of the stopword and stemming technique in Urdu IR. They found that the stopword removal and stemming technique improve the performance of an IR system. Sahu and Pal (2023) built a text collection for Sanskrit and evaluated different stemming strategies in the text analysis domain. They observed that different pre-processing strategies improve the performance of the Sanskrit NLP and IR domain.

Since a rule-based stemmer could not be used in different languages, various researchers presented language-independent stemmers (Xu and Croft (1998), Goldsmith (2001), Majumder et al. (2007)) and evaluated their effectiveness in different lan-

[5] forum for information retrieval evaluation

guages. They showed that language-independent stemmers offer comparable performance to rule-based stemmers in the IR domain. Mayfield and McNamee (2003) proposed an n-gram-based stemming technique in European languages. They observed that 4-gram provides the best performance in European languages. The major drawback of the n-gram approach is the size of the inverted index. This approach substantially expands the index size, which increases query processing time. The 4-gram model takes ten times more processing time than word-based retrieval. Buckley et al. (1995) observed that without knowledge of a language, an excellent stemmer could be constructed by analyzing the lexicon and most common suffixes. They proposed a stemmer in the Spanish text by observing the lexicographical similarities between the words.

Based on the above findings, we conclude that stemming improves retrieval effectiveness in European, Asian, and South Asian languages. However, the effect of language-independent stemming strategies in Indian languages has been less explored. This work explores the effect of language-independent stemming strategies in low-resource Indian languages IR. These findings may be helpful for other languages rich in morphology. Our evaluation strategy is in line with the earlier work of Majumder et al. (2007), Paik and Parui (2011), Paik et al. (2011b), Paik et al. (2011a), Silvello et al. (2018). In particular, we evaluate the following language-independent stemmers in Indian languages.

3 Different stemming approaches

In recent years, language-independent stemmers have performed similarly to rule-based stemmers in different languages. The primary benefit of a language-independent stemmer is that it does not require any linguistic knowledge to implement. Hence, we evaluated the following language-independent stemmers in low-resource languages from an IR perspective.

3.1 Yet Another Suffix Stripper (YASS)

Majumder et al. (2007) proposed a clustering-based stemmer for morphologically rich low-resource languages. We implemented the stemming technique using the algorithm 1.

Algorithm 1 Yet another suffix stripper

1. Word (W) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word (W') $\leftarrow$ list of stemmed words
3. They define four types of string distance measure D_1, D_2, D_3 and D_4 for clustering the lexicon
4. We use D_3 as string distance measure for clustering the lexicon because it yields the least significant difference in retrieval performance at different threshold values
5. For given two words in the lexicon X and Y, if 'x' is the maximum length of X and Y and 'y' is an index of the first mismatch between X and Y, then D_3 is defined as : $D_3(X, Y) = \frac{x-y+1}{y} * \sum_{i=y}^{x} \frac{1}{2^{i-x}}$
6. To identify morphologically similar terms, a complete linkage clustering algorithm is used
7. During clustering, we experimented with different threshold values (θ), and the best MAP score obtained at a particular threshold value is noted down in Section 6
8. Compared the MAP score of D_3 based stemming approach with baseline (no stemming approach).

3.2 Fast corpus-based Stemmer (FCB)

Paik and Parui (2011) proposed a statistical stemmer that uses the suffix frequency to produce a root word. We implement the stemming strategy using the algorithm 2.

Algorithm 2 Fast corpus-based stemmer

1. Word (W) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word (W') $\leftarrow$ list of stemmed words
3. Based on the common prefix and potential suffix information, they categorize the words into the k-equivalence class
4. If the suffix frequency exceeds a cut-off threshold (α), it is referred to as a potential suffix (β)
5. The longest common prefix of each equivalence class is treated as a possible stem or root word for the class
6. The ratio of the size of the potential class to the size of the generated class determines the strength of the prefix
7. If the evaluated ratio exceeds a specified threshold δ, the longest prefix of the class is treated as a valid stem. Otherwise, a better stem is found by applying the above process iteratively
8. Compared the MAP score of FCB V-1 based stemming approach with baseline (no stemming technique).

Paik and Parui (2011) shows that k_1=3 and k_2=2 provide best retrieval performance in the Indian languages IR. Hence, in this study, we experimented with different values of k_1, k_2 and δ. Our evaluation technique aligns with the previous work of Silvello et al. (2018).

3.3 Co-occurrence based stemmer (SNS)

Paik et al. (2011b) proposed a statistical stemmer based on the co-occurrence statistics in the corpus. We implemented the SNS stemmer using the algorithm 3.

Algorithm 3 A co-occurrence based stemmer

1. Word (W) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word (W') $\leftarrow$ list of stemmed words
3. Determine the co-occurrence strength of word pairs
4. Using neighbours re-calculate the co-occurrence strength
5. The words are grouped according to their newly determined co-occurrence strength. The co-occurrence of two words, a and b, is defined as: $CO(a,b) = \sum_{d \in C} min(tf_{\text{a,d}}, tf_{\text{b,d}})$
   where d represents document and $tf_{\text{p,d}}$ the term frequency of term p in d
6. The words are now mapped into a weighted undirected graph, in which each word says w_1 and w_2 are represented as a node, and they are connected by an edge (w_1, w_2) with weight CO(w_1, w_2) if it satisfies at least one of these two conditions:
   (i) CO(w_1, w_2) > 0; and
   (ii) Length of common prefix between w_1 and w_2 is at least L_1, (Here L_1=3) along with the suffixes which are suffix of more than one co-occurring words after removal of longest common prefix larger than L_2 (Here $L_2 > 5$)
7. If both the words co-occur with other words, then we re-calculate the co-occurrence strength by the following equation
   $RCO(a,b) = CO(a,b) + \sum_{c \in N_{\text{a,b}}} min(CO(a,c), CO(c,b)) * 0.5$
   Where $N_{\text{a,b}}$ denotes the set of common neighbours of a and b
8. The strong edges will be kept, while the weak edges will be removed. The stem is the longest prefix among the connected components of the graph
9. Compared the MAP score of the co-occurrence-based stemming approach with the baseline (no stemming approach).

3.4 Graph based (GRAS) stemmer

Paik et al. (2011a) presented a statistical stemmer specifically for highly inflectional languages. The GRAS stemmer is used in different text analysis tasks because of less computational effort, effectiveness in retrieval, and language-independent nature. We implemented the GRAS stemmer using the algorithm 4.

Algorithm 4 GRAS Stemmer

1. Word (W) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word (W$^{'}$) $\leftarrow$ list of stemmed words
3. GRAS identifies the word partitions sharing using an L-long prefix. Where L is the average word length of the language
4. We identify and save the η-frequent suffix pairings for each common prefix. A large value of η causes the omission of many valid suffix pairs; hence a low η value is safer. Here, we use η =1 to avoid omission of valid suffix pairs
5. A graph is constructed, where each node represents a word, and each edge represents the morphological link between two words
6. The words are divided into many equivalence groups. The morphological relationship between word and pivot is determined by the cohesion value (δ)
7. If a large number of edges are connected to a node, then it is treated as a pivot node or stems
8. Compared the MAP score of the graph-based stemming approach with the baseline (no stemming technique).

We also evaluate language-independent indexing strategies, that is, Trunc-n (truncation of the first n letters). The trunc-4 truncates the first four letters (e.g., 'educated' provides 'educ'). The best MAP score obtained by stemming techniques for different languages with different parameters and 'n' values is shown in Table 1.

Table 1: Different parameters used for stemming method evaluation

	YASS	FCB V-1	SNS	GRAS	Trunc-n
Hindi	$\theta = 1.5$	k_1=4, k_2=2, $\delta = 0.7$	L_1=4 L_2=6	L=6 $\alpha = 4$	5
Gujarati	$\theta = 0.6$	k_1=4, k_2=2, $\delta = 0.6$	L_1=5, L_2=7	L=7 $\alpha = 4$	5
English	$\theta = 1.55$	k_1=7, k_2=2, $\delta = 0.6$	L_1=3 L_2=5	L=8 $\alpha = 6$	6

where

θ, α, δ : Threshold taken by different stemmers
k_1 : Initial prefix length
k_2 : Final prefix length
L : Average word length of the language
L_1, L_2 : Length of common prefix

4 Information Retrieval Framework

We used different document weighting and ranking models supported by Terrier [6] retrieval system to evaluate the effectiveness of stemming methods. Terrier supports various IR models, such as probabilistic, DFR-based, and language models. This experiment used probabilistic retrieval models (BM25 and TF-IDF), DFR-based retrieval models (BB2, InL2, IFB2), and the Hiemstra language model.

5 Test Collection

We experimented with different Indian language test collections. The test collections are part of the FIRE [7] evaluation campaign. The collections mainly consist of news articles extracted from different archives. Table 2 shows the statistics of different test collections. In the collections, both topics and documents use the UTF-8 encoding system. This experiment considers only the query's title (T) section.

Table 2: Shows the statistics of the text collection

Collection	Size	Number of documents	Number of queries
Hindi	1.3 GB	331608	50
Gujarati	2.2 GB	313163	50
English	1.1 GB	392577	50

[6] http://terrier.org/
[7] http://fire.irsi.res.in/fire/static/data

Table 3: Retrieval results in Hindi 2011 text collection (50 T queries)

	↓ Parameter — R.M. →	BM25	TF-IDF	BB2	InL2	IFB2	LM
Base line	MAP	0.4444	0.4455	0.3746	0.3909	0.3724	0.405
	Rel.Ret.	1683	1679	1659	1654	1664	1667
YASS	MAP	0.442	0.446	0.3773	0.3785	0.3675	0.3989
	Rel.Ret.	1764	1762	1729	1735	1731	1730
FCB V-1	MAP	0.4476	0.4488	0.3767	0.3939	0.3746	0.4081
	Rel.Ret.	1725	1720	1703	1700	1708	1708
SNS	MAP	0.4483	**0.4588**	**0.3798**	**0.3945**	**0.3808**	**0.4092**
	Rel.Ret.	1745	1742	1732	1724	1728	1712
GRAS	MAP	**0.4495**	0.4505	0.3692	0.3815	0.3661	0.3995
	Rel.Ret.	1759	1755	1735	1741	1740	1736
Trunc-n	MAP	0.4543	0.4561	0.376	0.3879	0.3729	0.409
	Rel.Ret.	1749	1744	1733	1725	1735	1706

6 Evaluation

In the first set of experiments, we evaluate the effect of different language-independent stemming techniques in Indian languages IR. In Hindi, MAP, and the relevant documents retrieved in the baseline and different language-independent stemming methods are shown in Table 3. We conduct similar experiments for the other Indian languages: Gujarati and English, as shown in Table 4, and 5 respectively. It is observed that different stemming techniques improve MAP scores in Indian languages IR. The best performance by a stemming approach is shown in boldface. The SNS stemmer provides the best MAP score in Hindi and Gujarati. However, the GRAS stemmer provides the best MAP score in English. The trunc-n-based indexing strategy offers similar performance to the SNS stemmer in Hindi and Gujarati. Moreover, the GRAS stemmer provides comparable performance in English. During the evaluation of different retrieval models, we observed that the probabilistic retrieval models (BM25 and TF-IDF) give the best retrieval performance in Hindi, Gujarati and English. The DFR-based retrieval models (BB2, InL2, and IFB2) exhibit poor performance in Indian languages IR.

We perform a query-by-query analysis to get more insight into the effect of stemming in Indian languages. Here, we consider the best retrieval models and stemming approaches for Indian languages. We consider the SNS stemmer for Hindi and Gujarati languages and the GRAS stemmer for English. In closer observation, we found that stemming improves performance for 35 topics in Hindi and reduces performance for 15 topics. The

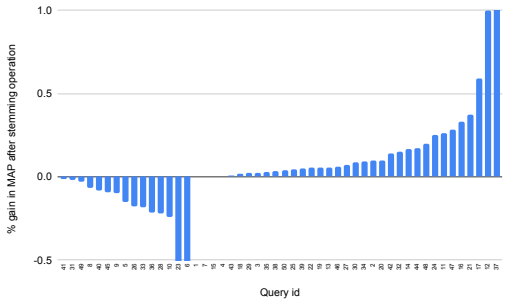

Figure 1: A query by query evaluation in Hindi by SNS stemmer in TF-IDF model

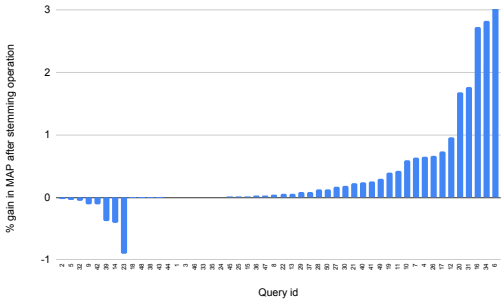

Figure 2: A query by query evaluation in Gujarati by SNS stemmer in InL2 model

performance of each query is shown in Fig. 1. Likewise, in Gujarati and English, stemming improves performance in 38, and 32 topics, respectively, and reduces performance in 8, and 12 topics. The percentage changes in performance due to stemming at the per-query level are shown in Fig 2, and 3. From query-by-query analysis, we also observed that the stemming performs better in Gujarati, and English than in Hindi.

During the experimentation of different

Table 4: Retrieval results in Gujarati 2011 text collection (50 T queries)

	↓ Parameter — R.M. →	BM25	TF-IDF	BB2	InL2	IFB2	LM
Base line	MAP	0.24	0.2399	0.2041	0.1992	0.2021	0.2095
	Rel.Ret.	1315	1308	1278	1270	1274	1289
YASS	MAP	0.2464	0.2463	0.2116	0.2057	0.2077	0.2137
	Rel.Ret.	1320	1311	1280	1279	1276	1287
FCB V-1	MAP	0.2423	0.2404	0.2056	0.1998	0.2031	0.2105
	Rel.Ret.	1337	1331	1292	1290	1296	1313
SNS	MAP	**0.2647**	**0.2643**	**0.2385**	**0.2406**	**0.2342**	**0.2335**
	Rel.Ret.	1359	1357	1343	1325	1336	1338
GRAS	MAP	0.2443	0.2439	0.2125	0.2167	0.2105	0.2184
	Rel.Ret.	1349	1342	1321	1305	1314	1329
Trunc-n	MAP	0.2579	0.2578	0.2282	0.2331	0.2246	0.2282
	Rel.Ret.	1360	1356	1342	1330	1333	1341

Table 5: Retrieval results in English 2011 text collection (50 T queries)

	↓ Parameter — R.M. →	BM25	TF-IDF	BB2	InL2	IFB2	LM
Base line	MAP	0.2975	0.2981	0.2686	0.2633	0.2615	0.2543
	Rel.Ret.	2236	2232	2210	2204	2210	2182
YASS	MAP	0.3122	0.3133	0.2837	0.2769	0.2745	0.2652
	Rel.Ret.	2337	2337	2338	2312	2338	2278
FCB V-1	MAP	0.3012	0.302	0.2723	0.2662	0.2651	0.257
	Rel.Ret.	2278	2277	2268	2249	2269	2221
SNS	MAP	0.3068	0.3065	0.2753	0.2709	0.2734	0.2611
	Rel.Ret.	2278	2279	2250	2240	2249	2224
GRAS	MAP	**0.3145**	**0.3155**	**0.2849**	**0.2796**	**0.2763**	**0.2661**
	Rel.Ret.	2310	2311	2294	2280	2290	2248
Trunc-n	MAP	0.3155	0.3164	0.2858	0.2818	0.2772	0.267
	Rel.Ret.	2309	2310	2295	2277	2290	2247

language-independent stemmers (shown in Table 3, 4, and 5), we observe that stemming improves retrieval performance in different Indian languages IR. On closer observation, we found that the effect of stemming varies in different Indian languages. The SNS stemmer performs best and improves a MAP score of 2.98% in Hindi, 20.78% in Gujarati IR. Similarly, the GRAS stemmer performs best and improves a MAP score of 5.83% in English IR. Among the different stemming techniques experimented with, the GRAS stemmer required less computational effort and performed best in different Indian languages. We conclude that the language-independent stemmer improves retrieval performance in different Indian languages IR. This observation is similar to the findings in other Indian and European languages by (Majumder et al., 2007) and (Paik and Parui, 2011).

7 Conclusion

Stemming is an essential preprocessing step in the IR system. The above experiments show that stemming improves retrieval performance in different Indian languages compared to the baseline approach (no stemming). Different stemming techniques perform best in Gujarati and English languages. However, the stemming technique provides a relatively poor performance in Hindi. The SNS stemmer performs best in Hindi and Gujarati, whereas the GRAS stemmer performs best in English. The trunc-n-based indexing strategy performs similarly to the best-stemming approaches in different Indian languages. During the evaluation of the retrieval models, we observe that the probabilistic retrieval models (BM25 and TF-IDF) perform best in Hindi, Gujarati and English languages. The DFR-based retrieval models provide poor performance in different Indian languages.

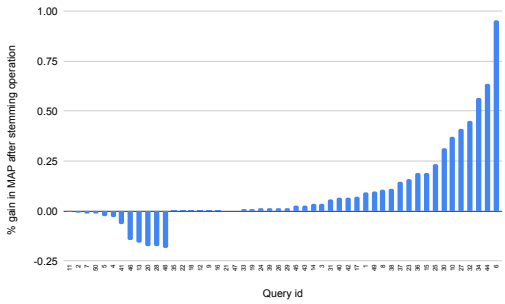

Figure 3: A query by query evaluation in English by GRAS stemmer in TF-IDF model

Although the effect of stemming is thoroughly investigated in the Indo-European language family, it is less explored in the Dravidian language family. India has significant native speakers in Dravidian languages such as Telugu, Tamil, Kannada, and Malayalam. So, it will be interesting to explore the effect of different stemming techniques in the Dravidian language family in the future. Moreover, one can also study the impact of different machine learning-based and deep learning-based stemming techniques in different Indian and European languages IR.

8 Acknowledgements

This work is supported by IIT (B.H.U), Varanasi. Moreover, the support and resources provided by the PARAM Shivay Facility under the National Supercomputing Mission, Government of India, at IIT (B.H.U), Varanasi, are gratefully acknowledged.

References

Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. 1995. New retrieval approaches using smart: Trec 4. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48.

John Dawson. 1974. Suffix removal and word conflation. *ALLC bulletin*, 2(3):33–46.

Ljiljana Dolamic and Jacques Savoy. 2010. Comparative study of indexing and search strategies for the hindi, marathi, and bengali languages. *ACM Transactions on Asian Language Information Processing (TALIP)*, 9(3):1–24.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

David A Hull. 1996. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84.

Julie Beth Lovins. 1968. Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, 11(1-2):22–31.

Prasenjit Majumder, Mandar Mitra, Swapan K Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. Yass: Yet another suffix stripper. *ACM transactions on information systems (TOIS)*, 25(4):18–es.

James Mayfield and Paul McNamee. 2003. Single n-gram stemming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 415–416.

Jiaul H Paik, Mandar Mitra, Swapan K Parui, and Kalervo Järvelin. 2011a. Gras: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 29(4):1–24.

Jiaul H Paik, Dipasree Pal, and Swapan K Parui. 2011b. A novel corpus-based stemming algorithm using co-occurrence statistics. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 863–872.

Jiaul H Paik and Swapan K Parui. 2011. A fast corpus-based stemmer. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10(2):1–16.

Carol Peters. 2008. *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, volume 5152. Springer Science & Business Media.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*.

Siba Sankar Sahu, Debrup Dutta, Sukomal Pal, and Imran Rasheed. 2023. Effect of stopwords and stemming techniques in urdu ir. *SN Computer Science*, 4(5):547.

Siba Sankar Sahu and Sukomal Pal. 2023. Building a text retrieval system for the sanskrit language: Exploring indexing, stemming, and searching issues. *Computer Speech & Language*, 81:101518.

Jacques Savoy. 2006. Light stemming approaches for the french, portuguese, german and hungarian languages. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1031–1035.

Gianmaria Silvello, Riccardo Bucco, Giulio Busato, Giacomo Fornari, Andrea Langeli, Alberto Purpura, Giacomo Rocco, Alessandro Tezza, and Maristella Agosti. 2018. Statistical stemmers: A reproducibility study. In *European Conference on Information Retrieval*, pages 385–397. Springer.

Jinxi Xu and W Bruce Croft. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)*, 16(1):61–81.