Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 233 (2024) 902–908

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024)

Evaluating the Effect of Preprocessing Tools for Marathi Text Retrieval

Harshali B. Patil[a], Ajay S. Patil[b]

[a]*Dr. Annasaheb G. D. Bendale M. M. Jalgaon 425001, India*
[b]*School of Computer Sciences, KBCNMU Jalgaon 425002, India*
[a]*patilharshalib@gmail.com*,[b]*ajaypatil.nmu@gmail.com*

Abstract

The dramatic growth of the e-content available on the Internet in non-English languages facilitates the researchers to develop tools and techniques for automated processing of these languages. Retrieving meaningful information from this massive data is a challenging task, hence information retrieval of non-English languages is gaining more focus since last decade. The use of pre-processing tools like: stemmers, stop-word removal, lemmatizers, etc. has proven highly effective for the task of Information Retrieval for many languages like: English, Arabic, Hindi, etc. The goal of this work is to propose a simple stemmer for Marathi language using suffix stripping mechanism and evaluates the impact of it along with stop-word removal for Marathi text retrieval process. The result shows that significant improvement is obtained in the terms of precision, r-precision, precision@10, and recall due to the use of proposed suffix stripper and stop-word removal tool for Marathi text retrieval.

Keywords: Text retrieval;marathi;stemming;stop-word removal

1. Introduction

In this digital era, there is a massive amount of text data available on the Internet in several languages. Retrieving meaningful information from this huge data is a challenging task which includes several phases. Pre-processing techniques are found effective for Information Retrieval (IR) task. Stemmer is one of the important pre-processing components that are used in most of the natural language processing systems like information extraction, text mining, information retrieval, etc. Stemmers are used in information retrieval systems mainly for two reasons; one is that it reduces the size during indexing and the other is that it increases recall of the IR system. Stemming is a one of the important word normalization techniques that transforms the morphological variations of the word to their common representation called as stem. Most of the work related to English and non-English stemmer development has been

10.1016/j.procs.2024.03.279

already carried out and many state-of-the art techniques have been used for development of stemmers for these languages. Techniques like rule-based, clustering-based, corpus-based, hybrid, etc. have been used for stemmer development. The stemmer developed for information retrieval should be fast and efficient; hence we have developed a simple suffix stripper for Marathi language. Marathi is the official language of Maharashtra state and co-official language of Goa. Marathi is highly inflectional language hence word normalization technique like stemming can be effective for Marathi text retrieval.

Stop-word removal is also found effective for IR process for several languages. Stop-words are the high frequency words that don't carries any meaningful information related to the document but they are used several times while document creation and query construction. For instance, "is, they, a, an, the", etc. are the stop-words of English language. This paper discusses the development of suffix stripper for Marathi language and stop-word list construction along with the impact of both of these tools on Marathi text retrieval process. The rest of the paper is organized as: Section 2 presents the related work carried out for text information retrieval. Section 3 presents the proposed simple suffix stripper, and stop-word list development process. Section 4 describes the text search results, while section 5 concludes the paper.

2. Related work

The work related to stop-word list and stemmer development has a long history and their impact on text retrieval is also studied for several languages. Light stemming approach developed by Aljlayl [1] improved the retrieval effectiveness on Arabic search. Braschler et. al [2] studied the effectiveness of stemming and decompounding for German text retrieval and found that stemming was satisfactory even when simple approach was used. The authors carefully designed decompounding which boosted the performance. Word normalization using stemming, lemmatization and decompounding in mono-and bilingual IR has been studied by Airio [3] and concluded that normalization tools did not remarkably improve the retrieval result of monolingual English run, but in non-english runs they gave significantly better results. Orengo et al [4] presented a study on the use of stemming for monolingual ad-hoc Portuguese information retrieval and showed significant improvement in terms of mean average precision (MAP), precision at ten, the number of relevant retrieval with their lighter stemmer. Non-english web search: an evaluation of indexing and searching the Greek web has been done by Efthimiadis et al. and concluded that although the global search engines outperformed the Greek engines there is much room for improving their performance in both retrieval effectiveness and coverage [5].

In the context of Indian languages very less work has been found. Pande et al evaluated the effect of stemming and stop-word removal on Hindi text retrieval and the experimental investigation suggested that stop word removal improved retrieval significantly, however, a small drop in retrieval precision is obtained with all the stemmers [6]. Almeida and Bhattacharyya used n-gram as a word normalization technique and obtained a MAP of 0.4483 for information retrieval [7]. Few more stemmers have been developed for Marathi language like: Majgaonker & Siddiqui[8], Husain [9] Patil & Patil[10], Patil et.al.[11], Patil & Patil[12], Giri et.al.[13] Kadam et.al.[14]; however, their impact on the information retrieval process has not been reported.

3. Proposed pre-processing tools

This section describes the proposed preprocessing tools: stemmer and stop-word list along with the text retrieval experiment. Stop-words are the high frequency words which carry negligible information related to the document; hence they are omitted during the process of information retrieval. Here the development processes of Marathi stop-word list, stemmer along with the retrieval strategy is discussed.

3.1. Proposed Marathi stop-word list

The development of stop-word list required large corpus, hence FIRE corpus is used for development of proposed stop-word list. This corpus consists of 99275 documents of Marathi new papers *eSakal* and *Maharashtra Times* spanning the year 2004 to 2007. All the documents are encoded in UTF-8 format and the total collection is of 485MB.

The guidelines described by Fox [15] are adopted for development of Marathi stop-word list. The Fox guidelines are based on term frequency. The development process of proposed stop-word list is as given below:

- Initially all the word forms appearing in the corpora are sorted according to their frequency of occurrence and then some topmost terms are extracted.
- Then the list obtained in above step is inspected to remove all numbers, nouns and adjectives.
- Finally, some stop-words are added, even if they didn't appear in the topmost frequent words.

The proposed stop-word list consists of 218 terms. The partial proposed Marathi stop-word list is presented in table 1.

Table 1. Stop-word list

आणि	नको	होत	झाली	परतु
आता	नये	आहेत	याचा	मध्ये
आदी	मला	आहे	याची	मात्र
आपण	याच	होता	याचे	याचा

3.2. Proposed suffix-stripper

The simple suffix stripper for Marathi language stemming is developed using the mechanism of order classes of suffixes. After observing the corpus terms, it is found that four order-classes: vowel signs, case markers, postpositions and verb endings are existing among Marathi suffixes. The first order-class i.e. the class that appear immediately after stem consist of vowel Signes (VS) / plain suffixes (PS). These suffixes are used to convert root / stem into *samanyaroop* which is done before attaching any further suffixes, or it may be used for denoting the number, gender, etc. The next order-class consists of verb endings (VE) while the third order class consists of postpositions (PP) and the last order-class consists of case markers (CM) suffixes. For development of simple Marathi stemmer initially the suffixes are learned from the popular Marathi grammar books [16]. For obtaining the stem of word based on simple stemmer the algorithm 1 is followed. An iterative stemming algorithm simply removes suffixes in each order-class one at a time, starting at the end of a word and working backward to its beginning. Removing more than one suffixes belong to single class degrades the performance of stemming; therefore, more than one match between the same order-class is not handled in the algorithm. The rules for removing a suffix are given in the form of "if" conditions. The stemming process uses 249 total suffixes. Table 2 depicts the information related to some of the suffixes present in each class along with examples.

Table 2. Suffix classes

Sr. No	Class	Suffixes	Examples
1	CM	चा, चे, ने	अलीकचा, अल्पकाळच, अवंतिकेने
2	PP	वरील, साठी	अल्पबचतीवरील, अल्पसंख्यकांसाठी
3	VE	ते, णे, वा	अवघडते, अवतरणे, अवतरावा
4	PS	ी, े, ा	अवकाशी, गावे, अवगुणी

Algorithm 1 : Marathi Suffix Stripper

Input : File *f* consisting of news article
CM - Set of Marathi case markers,
PP - Set of Marathi prepositions,

PS - Set of Marathi plain suffixes,
VE - Set of verb endings
Output: Stemmed file *f'*

Begin
SV = ø // SV is use to store the stems of the words
Read *f*
Tokenize *f* and populate token vector *TV*
For Every token *t* in *TV* do
If *t* ends with suffix *x* such that *x* ϵ *CM* // step 1
t = *t* - *x*
Endif
If *t* ends with suffix *x* such that *x* ϵ *PP* // step 2
t = *t* - *x*
Endif
If *t* ends with suffix *x* such that *x* ϵ *VE* // step 3
t = *t* - *x*
Endif
If *t* ends with suffix *x* such that *x* ϵ *PS* // step 4
t = *t* - *x*
Endif
Add *t* to *SV*
Endfor
Write *SV* in output file *f'*
End

---

4. Results

For measuring the effectiveness of an IR system in a standard way, a test collection consisting of following things is needed:

- A document collection- For evaluating the impact of proposed pre-processing tools in the task of Marathi text retrieval we have created our own test corpus. This corpus is developed from online archive of *Daily Sakal* newspaper. Total 500 news articles of size 2.26 MB are collected for forming the test-data. All the documents are encoded in the Unicode. Total numbers of terms present in those documents are 119432 among these 27111 unique terms are present. The average number of terms per document is 238.86.
- A test suite of information needs, expressible as queries. Based on the document collection 15 queries are constructed related to each topic. More than 10 relevant documents per query are present in the corpus. The search query consists of maximum 8 words and minimum 3 words.
- A set of relevance judgment- Standard measures like precision, recall, along with r-precision and precision@10 is used for measuring the impact of proposed preprocessing techniques. Fig. 1 shows the processes involved in evaluating preprocessing tools for Marathi text retrieval.

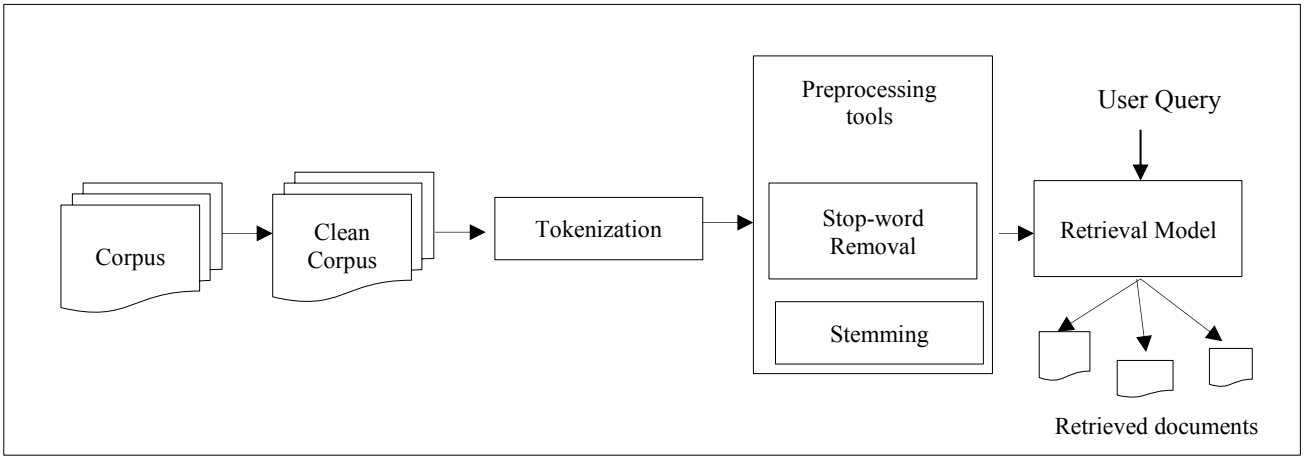

Fig. 1 : Process involved in evaluating preprocessing tools

Following table 3 reports the results obtained for Marathi text retrieval process using proposed pre-processing tools. In baseline, the clean corpus is used for searching the relevant document with respect to given search query. The documents as well as queries are not pre-processed. The second part of the table consists of the retrieval results when suffix stripper and stop-word removal is performed for queries as well as documents.

Table 3. Retrieval results obtained for baseline and with the use of pre-processing tools

Query no	Baseline results				Results obtained after pe-processing			
	P	p@10	R-precision	R	P	p@10	R-precision	R
1	0.1095	0.9	0.577	0.9230	0.2181	0.8	0.6000	0.9230
2	0.2545	1.0	0.929	1.0000	0.4642	1.0	0.9000	0.9285
3	0.1381	0.8	0.468	0.9787	0.73015	1.0	0.9000	0.9787
4	0.0742	0.9	0.769	1.0000	0.4137	0.9	0.8000	0.9230
5	0.7222	0.8	0.722	0.3714	0.1391	1.0	0.4900	0.7714
6	0.0454	0.1	0.125	0.4375	0.2500	0.8	0.6300	0.8750
7	0.0714	0.8	0.500	1.0000	0.4000	0.8	0.7778	1.0000
8	0.4125	0.8	0.600	0.6346	0.1284	0.6	0.4040	0.8846
9	0.7857	0.9	0.790	0.5789	0.2602	0.7	0.7900	1.0000
10	0.3333	0.6	0.545	0.8181	0.3793	0.6	0.6400	1.0000
11	0.2500	0.3	0.455	0.8181	0.1666	0.9	0.9090	1.0000
12	0.0729	0.5	0.500	0.8750	0.2295	0.7	0.5000	0.8750
13	0.1969	0.8	0.800	1.0000	0.2549	0.9	0.8500	1.0000
14	0.0376	0.5	0.545	1.0000	0.3055	0.9	0.9100	1.0000
15	0.1283	0.6	0.279	0.8837	0.8076	0.8	0.8100	0.4883

From table 3 it is observed that the precision and recall values are lower for most of the queries in baseline. This happens due to the presence of stop-words in the queries and documents, and the ambiguities present in the document text. Some documents with less similarity value are also present in the retrieval results due to the matching of stop-word in query and its existence in the corpus. Some relevant documents are missed during retrieval because these documents contain different variations of the query terms rather than the one that is used in the search query.

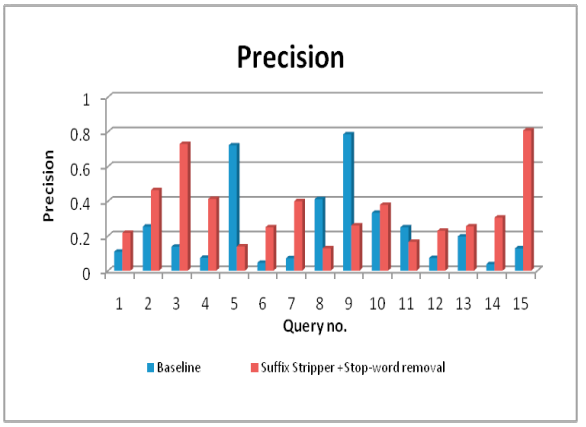

Fig. 2 (a): Precision obtained with baseline and pre-processing

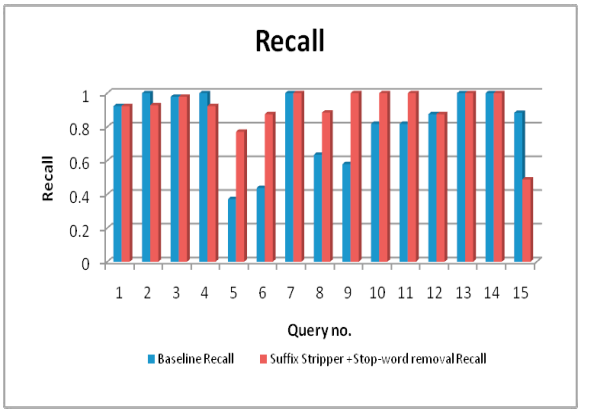

Fig. 2 (b): Recall obtained with baseline and pre-processing

From Fig. 2 (a) & (b) it is clearly observed that the precision value is increased for 11 queries out of 15, and recall value is increased or similar for 12 queries due to use of proposed pre-processing tools. This happens due to the fact that each match between the query and document is based on the meaningful term. Some relevant documents that are missed during baseline retrieval, due to use of different morphological variation of the same term in query and documents are also retrieved when suffix stripper is used as pre-processing tool for the queries and the documents.

5. Conclusion

This paper presents the development of pre-processing tools: suffix stripper and stop-word removal for Marathi text retrieval process. The impact of proposed suffix stripper along with stop-word removal is also studied for Marathi text retrieval process. The experimental result shows significant improvement in terms of precision, recall for many queries due to use of proposed pre-processing tools. The use of simple suffix stripper and stop-word removal boosts the performance of Marathi text retrieval system. In future, the applicability of these tools can be checked for other natural language processing applications as well as the impact of other types of stemmers will also be evaluated for Marathi text retrieval process.

Acknowledgements

The authors are very much thankful to the Forum for Information Retrieval Evaluation for providing the Marathi corpus which has been used in this research work.

References

[1] Aljlayl, Mohammed, and Ophir Frieder (2002) "On Arabic search: improving the retrieval effectiveness via a light stemming approach." In Proceedings of the eleventh international conference on Information and knowledge management, 340-347.

[2] Braschler, Martin, and Bärbel Ripplinger (2004) "How effective is stemming and decompounding for German text retrieval?" *Information Retrieval* **7**: 291-316.

[3] Airio, Eija (2006) "Word normalization and decompounding in mono-and bilingual IR." *Information Retrieval* **9**: 249-271.

[4] Orengo, Viviane Moreira, Luciana S. Buriol, and Alexandre Ramos Coelho (2006) "A study on the use of stemming for monolingual ad-hoc Portuguese information retrieval." In Workshop of the Cross-Language Evaluation Forum for European Languages, pp. 91-98. Berlin, Heidelberg: Springer Berlin Heidelberg.

[5] Efthimiadis, Efthimis N., Nicos Malevris, Apostolos Kousaridas, Alexandra Lepeniotou, and Nikos Loutas (2012) "Non-english web search: an evaluation of indexing and searching the Greek web." *Information Retrieval* **12**: 352-379.

[6] Pandey, Amaresh Kumar, and Tanvver J. Siddiqui(2009) "Evaluating effect of stemming and stop-word removal on Hindi text retrieval." In Proceedings of the First International Conference on Intelligent Human Computer Interaction: (IHCI 2009) January 20–23, 2009 Organized by the Indian Institute of Information Technology, Allahabad, India, 316-326.

[7] Almeida, Ashish, and Pushpak Bhattacharyya (2010) "Experiments in N-gram based indexing and retrieval in Marathi." In Forum for Information Retrieval Evaluation. 2010.

[8] Majgaonker, Mudassar M., and Tanveer J. Siddiqui (2010) "Discovering suffixes: A Case Study for Marathi.", *International Journal on Computer Science and Engineering* **2(8)**: 2716–2720.

[9] Husain, Mohd Shahid (2012) "An unsupervised approach to develop stemmer." *International Journal on Natural Language Computing (IJNLC)* **1(2)**: 15-23.

[10] Patil, Harshali B., and Ajay S. Patil (2017) "MarS: a rule-based stemmer for morphologically rich language Marathi." In 2017 international conference on computer, communications and electronics (Comptelix), 580-584.

[11] Patil, Harshali B., Neelima T. Mhaske, and Ajay S. Patil (2018) "Design and development of a dictionary based stemmer for Marathi language." In Smart and Innovative Trends in Next Generation Computing Technologies: Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part I 3,769-777.

[12] Patil, Harshali B., and Ajay S. Patil (2020) "A hybrid stemmer for the affix stacking language: Marathi." In Computing in Engineering and Technology: Proceedings of ICCET 2019,441-449.

[13] Giri, Virat (2021) "MTStemmer: A multilevel stemmer for effective word pre-processing in Marathi." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12(2): 1885-1894.

[14] Vaishali Kadam, P., B. Kalpana Khandale, and C. Namrata Mahender (2022) "Design and development of marathi word stemmer." In Proceedings of Second International Conference on Advances in Computer Engineering and Communication Systems:35-48.

[15] C. Fox, "A stop list for general text." *In ACM SIGIR Forum*, vol. 24, no. 1-2, pp. 19-21, ACM 1989.

[16] Burgess, Ebenezer (1854) “Grammar of the Marathi language”, American Mission Press.