International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-05, Issue-02, May 2019

Performance Analysis of Proposed Language Independent Stemmer

Dr. M.Kasthuri,

Asst. Professor, Dept. of Com. Science, Bishop Heber College (Autonomous), Tiruchirappalli, Tamil Nadu, India. stephenbasilkasthuri@gmail.com

Abstract - Information Retrieval is an emerging discipline that involves methods, models and patterns to find the documents of an unstructured nature in dynamic environment. Search Engines are playing a major role in Information Retrieval Systems to identify the morphological variants of the language using Stemming. Stemming is an important pre-processing step in query-based systems such as IRS, Web Search Engine, Natural Language Processing, Big Data Analysis, etc. The purpose of stemming is to diminish different grammatical or word forms to a common base form. The main focus of the experiment study is to measure the time and performance of the language independent stemmer in terms of Query Throughput, Query Latency and Mean number of Words per Conflation class.

Keywords – Language Independent Stemmer, Stemming, Information Retrieval, Natural Language Processing

I. Introduction

Information is being made available online. English and European Languages basically dominated the web. However, the web is getting multi-lingual. Especially, there has been a huge increase in the amount of web information available in Indian and other Asian languages. Web document in a large number of Indian languages like Hindi, Urdu, Bengali, Oriya, Tamil, Telugu and Marathi is now available in the electronic form [1], [2], [3]. Information Retrieval Systems (IRS) play a vital role in providing access to this information. Many natural languages are inflected such as Tamil, Hindi, Bengali, Latin, Hebrew, etc. In such languages several words sharing the same morphological invariant or root word can be related to the same topic. Natural language texts typically contain many different morphological variants of a basic word.

There are several stemming algorithms developed for various languages especially morphologically rich languages like Tamil, Hindi, Bengali, etc in recent years. The stemmers can be applied for different languages to increase the searching efficiency and reducing the vocabulary size of the indexed files in information retrievals. The overall performance of the Information Retrieval System is increased after applying the stemming concept [4], [5]. Proposed Architecture for Language Independent Stemmer, System Implementation and Testing of Proposed Language Independent Stemmer published in [6], [7]. This research paper presents overall performance measurement of proposed language independent stemmer using Query Throughput (QT), Query Latency (QL) and Mean number of Words per Conflation class (MWC).

II. Review of Literature

Several studies have been conducted till recently to construct language independent stemmer in diversified applications. From the literature study, stemming approaches can be classified as manual and automatic [8]. Manual conflation is achieved during the searching time with right hand truncation by the user on query side but not on the documents side. Automatic conflation of stemmers is used for matching different forms of the same words with their root automatically. This approach performs on query side as well as documents side. Different types of automatic stemmers available as for examples rule-based, statistical, successor variety, table lookup, machine learning and hybrid are found in the literature [9]. The rule-based approach can be classified further into suffix removal and affix removal, and machine learning approach can also be further classified into supervised and unsupervised.

Robert et al. (2018) have proposed Experimental Analysis of Stemming on Jurisprudential Documents Retrieval. It is less aggressive stemmers, provided the best cost-benefit ratio, since they reduced the dimensionality of the data and improved the effectiveness of the information retrieval evaluation metrics in one of the analyzed collections [9]. However, this stemming research work mainly concentrated on jurisprudential documents.

Adege et al. (2017) have created a stemmer of Ge’ez language using rule-based approaches [13]. There are two approaches were followed such as affix removal and morphological analysis. The experimental result shows that, this research work provided with an accuracy of 82.42%. However, limited rules sets were created, which mainly affects the accuracy of the proposed stemmer. It requires

DOI : 10.35291/2454-9150.2019.0063

linguistic knowledge to generate rule set. Further, stemming errors like over-stemming and under-stemming problems were also observed from affix removal technique.

Zadeh et al. (2017) have proposed a new hybrid stemming method based on a combination of affix stripping and statistical techniques for Persian language [14]. The authors conducted a performance test on the proposed stemmer using two different data sets. The experimental result shows that encouraging results were obtained. However, rule-based affix stripping approach applied in this research work needss linguistic inspection and it is a time consuming. Additionally, if small snippets of documents are involved, then the approach will not provide effective results.

Patel et al. (2016) have proposed a lightweight stemmer for Gujarati language using supervised learning [15]. Initially the authors to create handcrafted rules for prefix and suffixes. Then these rules were checked using linguistic expert for the Gujarati Morphology. They evaluated the proposed algorithm with IRS and improved results were obtained. However, linguistic inspection needs lots of time. Additionally, the primitive linguistic knowledge of the language is more important, which not suitable for agglutinative as well as morphologically rich languages.

Ali et al. (2016) have proposed a rule-based stemming method for Urdu Text [16]. This approach has evaluated on Urdu headline news datasets. The proposed method provides 90% to 95 % accuracy. However, in order to develop this Urdu stemmer, generic stemming rules and stemming lists have been created in advance. Additionally, various grammar books and Urdu literature were used to generate a list of 60 prefix rules. It is a language dependent stemmer and linguistic inspection should be needed.

Nehar et al. (2015) have presented a stemmer for Arabic language using Trigram, Transducers and Rational Kernels approach [17]. The experimental result shows that stemming improves the quality of classifiers in terms of accuracy. However, the core limitation of this stemmer is that trigram approach requires large storage space and it is not a practical approach. The rational kernels classification is computationally expensive. In addition, complexity of this stemmer is heavy and it requires long computational time.

Deepamala et al. (2015) have proposed a stemmer for Kannada language with table lookup approach [18]. To improve the stemming performance, a stem list is generated manually and inserted into the table. However, the major problem in this stemmer is high complexity due to the adoption of various algorithms such as naïve bayes, lookup table approach and maximum entropy for classification and stemming. Large storage space is needed to insert each word and its morphological variants for the specific language. Finally, manual inspection is also needed to verify valid stem in the table.

Since more repositories are available for English and European languages on the web, there is a considerable amount of stemmers and lemmatizers available for them. Similarly, in the past few years, a wide range of information is being made available online for the Indian and other Asian languages by which many researchers have proposed language dependent stemmers for Tamil, Hindi, Bengali, Guajarati, etc. individually using rule-based approach [10], [11]. And most of the exiting search engines have not considered diacritics of the language for generating stem word [13]. However, there is no efficient and novel methodology for Language Independent Stemmer using Dynamic Programming to support different spoken languages. Hence, it has motivated to develop a new approach for creating multi-lingual stemmer to support all class spoken languages.

III. PERFORMANCE ANALYSIS

The data sample is collected from EMILLE [12] corpus and constructed four repositories namely English, French, Tamil and Hindi, arranged into conflation groups. The test data set-I has 1,858 English words, out of which 287 incorrect words are available. The test data set-II contains 1,858 French words, which contains 359 incorrect words. The test data set-III consists of 1,858 Tamil words, there are 356 incorrect words are identified. The test data set-IV contains 1,858 Hindi words, where 347 incorrect words are available. Similarly, ten different data samples are constructed with four data sets. Each data set has its own distinct words with various numbers of words collected from four different repositories. The maximum number of 2,70,674 distinct words are used in ten data samples. Table 5.1 shows that data sample1 with four data sets, chosen for the test.

Table 3 .1: Data Sample1 with Four Data Sets

Data Set	Total Number of Words	Total Number of Unique Words
Test Data set-I	1858	287
Test Data set-II	1858	359
Test Data set-III	1858	356
Test Data set-IV	1858	347

3.1. Query Throughput and Query Latency

Query throughput is the number of queries processed per second. Query latency is the execution time between issuing a query and receiving a response, which is measured in millisecond. Query throughput is measured using the execution time of a query i.e., dividing 1000 milliseconds by Query Latency. Relation between query throughput and latency is that high throughput can handle multiple queries and simultaneously query can produce high latency. The

DOI : 10.35291/2454-9150.2019.0063

proposed stemmer gives high throughput and can handle multiple queries. Simultaneously query latency for the proposed stemmer takes less time than other the existing stemmers. Table 3.2 shows Query Throughput and Query Latency for PLIS and other stemmers. Figure 3.1 shows the query latency for various stemmers.

Table 3.2: Query Throughput and Latency for Various Stemmers

Name of the Stemmer	Query Latency (in milliseconds)	Query Throughput (No.of Queries per Second)
Lovins Stemmer	954.3986345	1.05
Iterated Lovins Stemmer	967.4211452	1.03
Porter1 Stemmer	886.2011001	1.13
Porter2 Stemmer	897.1799336	1.11
Paice/Husk Stemmer	798.3698334	1.25
Fairwheather Stemmer	999.0887344	1.00
PECL/Porter French Stemmer	501.7911453	1.99
Damodharan Tamil Stemmer	634.5779266	1.58
Gupta Hindi Stemmer	988.2897334	1.01
PLIS	479.3783944	2.09

From the Query Throughput results, it is observed that the entire stemmers perform almost one query per second. However, the PLIS performs more than two queries per second. Figure 3.2 shows query throughput for various stemmers.

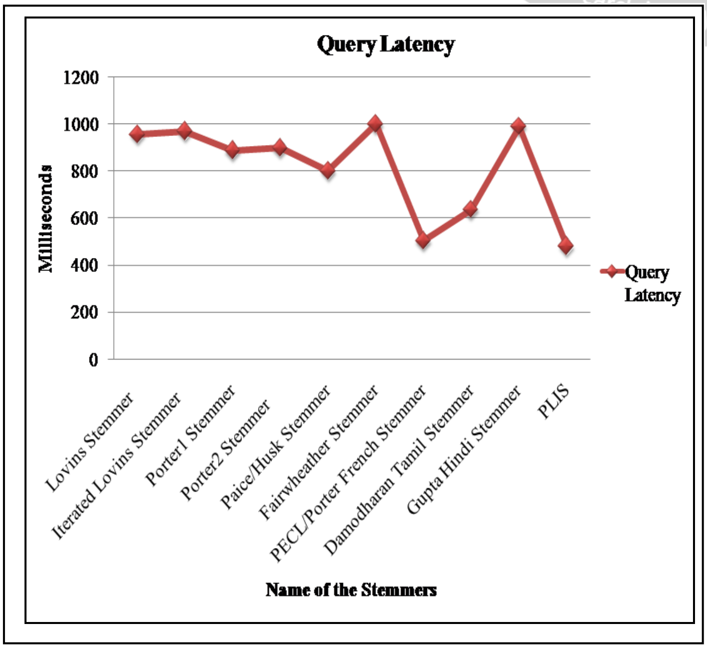

Figure 3.1: Query Latency for Various Stemmers

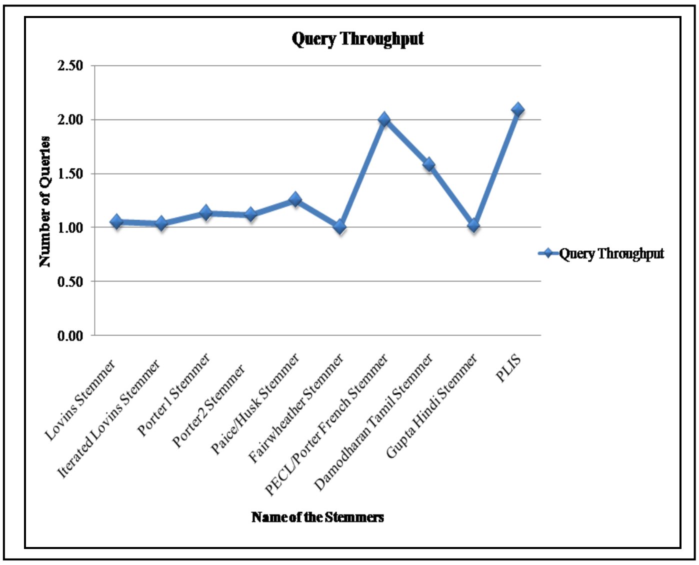

Figure 3.2: Query Throughput for Various Stemmers

3.2. Mean Number of Words per Conflation Class

Mean number of Words per Conflation (MWC) class is the important parameter, which is used to evaluate the performance of the proposed stemmer. It is the average size of the groups of words converted to a particular stem. Mean number of words can be calculated by dividing the number of unique words (N) with the number of unique stem (S) after stemming. Thus, if the words engineer, engineering and engineered are stemmed to the stem engineer, then the size of the conflation class would be 3. If the conflation of 1,000 different words (N) resulted in 240 distinct stems (S), then the mean number of words per conflation class would be 4.166667. Mean Number of Words per Conflation class (MWC) is calculated using the following equation (3.1).

$$\mathrm{MWC} = \frac{\mathrm{N}}{\mathrm{S}} \qquad \text{--------------------} \ (3.1)$$

N - Number of unique words before stemming

S - Number of unique stem after stemming

It is understood that the stronger stemmers will tend to have more words per conflation class. Table 3.3 shows the average MWC for PLIS including four languages and Table 3.4 shows MWC for PLIS and other existing stemmers. Figure 3.3 shows MWC for various stemmers.

DOI : 10.35291/2454-9150.2019.0063

Table 3.3: Average MWC for PLIS Including Four Languages

S. No	Number of unique words before stemming (N)	Number of unique stem after stemming (S)	Mean Number of Words = N/S
1	23674	10021	2.362438878
2	30000	12698	2.362576784
3	25000	10450	2.392344498
4	29000	12275	2.362525458
5	25000	10647	2.348079271
6	30000	12750	2.352941176
7	27000	11429	2.362411410
8	28000	11852	2.362470469
9	25000	10665	2.344116268
10	28000	11876	2.357696194
Average MWC			2.360760041

Table 3.4: MWC for PLIS and other Stemmers

Lovins Stemmer	Iterated Lovins Stemmer	Porter1 Stemmer	Porter2 Stemmer	Paice/ Husk Stemmer	Fairwheather Stemmer	PECL French Stemmer	Damodharan Tamil Stemmer	Gupta Hindi Stemmer	PLIS
2.30	11	2.08	2.16	2.32	2.32	1.96	1.97	1.99	2.36

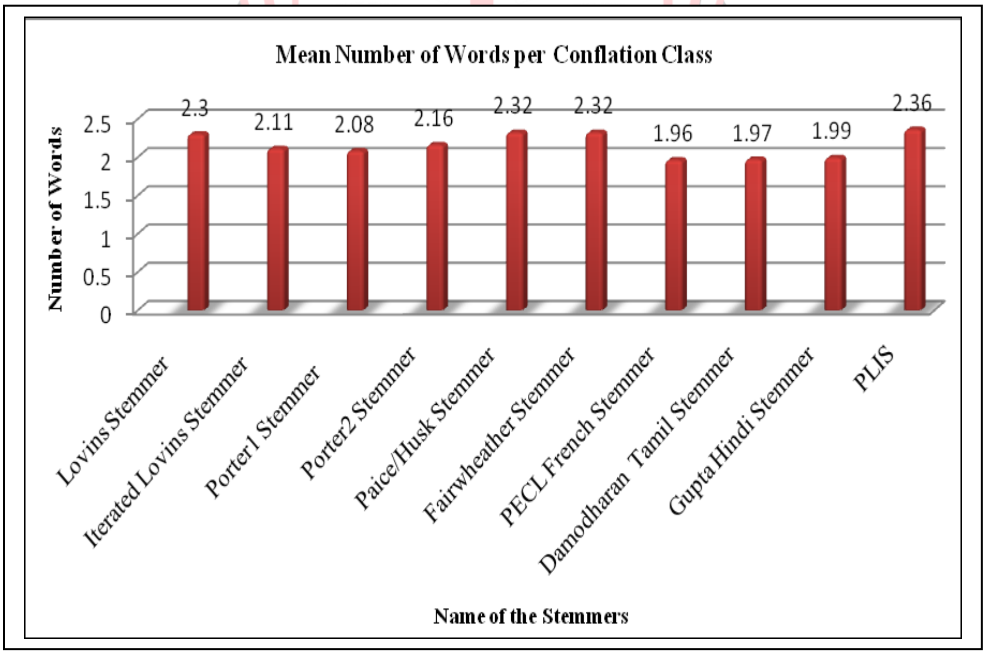

Figure 3.3: MWC for Various Stemmers

The scores obtained for Mean number of Words per Conflation class (MWC) of the proposed stemmer are positive. Therefore, the performance of the PLIS will be high. Average MWC for proposed language independent stemmer is tested against four languages. The average size of the words of a conflation group that are transformed to the same stem for the proposed language independent stemmer is 2.36.

IV. Conclusion

Stemming approaches in Information Retrieval Systems focus on increasing the retrieval performance, consuming less time but providing greater accuracy, strength and supporting multi-linguistic documents need more attention. In view of the above aspects, Proposed Language Independent Stemmer has developed. The performance of PLIS has analyzed in terms of Query throughput, Query Latency and Mean number of words per conflation class.

DOI : 10.35291/2454-9150.2019.0063

Query Latency score of PLIS is lower than the others stemmers and thus it has been concluded that processing time of PLIS takes less time than the other stemmers. Query Throughput is comparatively higher in PLIS than the other stemmers. But most of the stemmers give good result in Query throughput and Latency. From the table 3.4, Mean number of Words per Conflation class (MWC) obtained by all stemmer algorithms is above 2 characters. Therefore, from the above result it is observed that the PLIS provides better results compared to the existing stemmers based on its performance. The PLIS can be very well adopted in the applications that employ the stemming process. This proposed system can be extended to any Information Retrieval Systems, Search Engine, Natural Language Processing, Machine Translation, Computational Linguistic, Spell checker, Grammar Checker, Thesaurus, Word Frequency Counter, Document Summarizers, Indexers, etc. Strength and accuracy of PLIS will discuss in the next research paper.

ACKNOWLEDGMENT

I wholeheartedly thank University Grants Commission (UGC), Hyderabad for sanctioning the grant to complete the Minor Research Project entitled Language Independent Stemmer. I would like to express my sincere thanks to the Corpus Provider (EMILLE Corpus) who generously allowed me to download data from their website with free of cost.

REFERENCES

[1] Mohd. Shahid Husain, "An Unsupervised Approach to Develop Stemmer", *In: International Journal on Natural Language Computing (IJNLC),* ISSN: 2278-1307, Volume.1, Issue.2, India, 2012.

[2] Sajjad Ahmad Khan, Waqas Anwarl, Usama Ijaz Bajwa1, and Xuan Wang, "A Light Weight Stemmer for Urdu Language: A Scarce Resourced Language", *In: Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP),* pp.69–78, India, 2012.

[3] Dhabal Prasad Sethi, "Design of Lightweight Stemmer for Odia Derivational Suffixes", *In: International Journal of Advanced Research in Computer and Communication Engineering,* ISSN (Print): 2319-5940, ISSN (Online): 2278-1021, Volume.2, Issue.12, India, 2013.

[4] Kasthuri, M., and Dr. Britto Ramesh Kumar, S., "A Comprehensive Analyze of Stemming Algorithms for Indian and Non-Indian Languages", *In: International Journal of Computer Engineering and Applications (IJCEA),* ISSN: 2321-3469, Volume.7, Issue.3, pp.1-8, India, 2014.

[5] Kasthuri, M., and Dr. Britto Ramesh Kumar, S., "Multilingual Phonetic Based Stem Generation", *In: Second International Conference on Emerging Research in Computing, Information Communication and Applications (ERCICA-2014),* ISBN: 9789351072607, Volume.1, pp.437-442, ELSEVIER, India, 2014.

[6] Dr. M. Kasthur, "Proposed Architecture for Languag Independent Stemmer", *In: Journal of Emerging Technologies and Innovative Research (JETIR),* ISSN-2349-5162, Volume.5, Issue.10, pp: 943-948, India, 2018.

[7] Dr. M. Kasthur, "System Implementation and Testing of Proposed Language Independent Stemmer", *In: Journal International Journal of Computer Sciences and Engineering,* E-ISSN: 2347-2693, Volume.6, Issue.11, India 2018.

[8] Jyoti Katiyar, and Deepali Khatri, "Analysing the Process of Handing Difficult Keyword Queries over Databases", *In: International Journal of Informative & Futuristic Research,* ISSN (Online): 2347-1697, Volume.2, Issue.6, pp.1572-1580, India 2015.

[9] Robert A. N. de Oliveira and Methanias C. Junior "Experimental Analysis of Stemming on Jurisprudential Documents Retrieval", *In: Information,* Vol.9, Issue.28, doi:10.3390/ info9020028, pp.1-34, 27th January, Brazil, 2018.

[10] Safaa I Hajeer, Rasha Ismail, NagwaBadr and Mohamed Talba, "A New Stemming Algorithm for Efficient Information Retrieval Systems and Web Search Engines", *In: Multimedia Forensic and Security,* pp.117-135, Egypt, 2017.

[11] Gunadeep Chetia, Gopal Chandra Hazarika, "Pre-processing Phase of Automatic Text Summarization for the Assamese Language", *In: International Journal of Computer Sciences and Engineering,* Volume.6, Issue.10, pp.159-163, E-ISSN: 2347-2693, India, 2018.

[12] Emille Corpus, "Tamil and Hindi corpus", France, 2014. [http://catalog.elra.info/ product_info.php]

[13] Abebe Belay Adege, YibeltalChanieManie, "Designing a Stemmer for Ge'ez Text Using Rule Based Approach", *In: International Journal of Scientific & Engineering Research,* Vol.8, Issue.1, pp.1574-1578, ISSN: 2229-5518, Ethiopia, 2017.

[14] HosseinTaghi-Zadeh , Mohammad HadiSadreddini, Mohammad HasanDiyanati and Amir HosseinRasekh, "A new hybrid stemming method for persian language", *In: Digital Scholarship Humanities,* Volume.32, Issue.1, pp.209-221, Pakistan2017.

[15] Chandrakant D. Patel and Jayeshkumar M. Patel, "Improving a Lightweight Stemmer Language for Gujarati", *In: International Journal of Information Sciences and Techniques (IJIST,)* Volume.6, No.1/2, India, 2016.

[16] Mubashir Ali, Shehzad Khalid, HaneefSaleemi, WaheedIqbal, Armughan Ali and GhayurNaqvi, "A Rule based Stemming Method for Multilingual Urdu Text", *In: International Journal of Computer Applications,* ISSN: 0975 – 8887, Volume.134, No.8, Palistan 2016.

[17] AttiaNehar, DjelloulZiadi, and HaddaCherroun, "Rational Kernels for Arabic Stemming and Text Classification", *In: arXiv:1502.07504v1,* Algerie, 2015.

[18] Deepamala, N., and Ramakanth Kumar, P., "Kannada Stemmer and Its Effect on Kannada Documents Classification", *In: Computational Intelligence in Data Mining,* Volume.33, pp.75-85, Springer, India, 2015.

AUTHOR BRIEF INTRODUCTION

Dr. M.Kasthuri is working as an Assistant Professor in the Department of Computer Applications, Bishop Heber College, Tiruchirappalli, Tamil Nadu, India. She had completed her Doctorate of Philosophy in Computer Science in June 2017 at Bharathidasan University, Tiruchirappalli. She has published a number of National and International level research papers related to Web Mining and Stemming concepts. She has completed UGC sponsored Minor Research Project entitled as Language Independent Stemmer.

DOI : 10.35291/2454-9150.2019.0063