Comparative Study of Truncating and Statistical Stemming Algorithms

Sanaullah Memon[1]
Department of Information Technology, Shaheed Benazir Bhutto University, Shaheed Benazirabad Sindh, Pakistan

Prof. Dr. Ghulam Ali Mallah[2]
Department of Computer Science SALU, Khairpur, Sindh, Pakistan

K.N.Memon[3]
Department of Mathematics and Statistics, QUEST, Nawabshah, Sindh, Pakistan

AG Shaikh[4], Sunny K.Aasoori[5]
Department of BSRS QUEST Nawabshah Sindh, Pakistan

Faheem Ul Hussain Dehraj[6]
Department of Business Administration Shaheed Benazir Bhutto University Shaheed Benazirabad Sindh, Pakistan

***Abstract*—Search and indexing systems bear a significant quality called word stemming, is lump of content excavating requests, IR frameworks and natural language handling frameworks. The fundamental topic in the search and indexing through time is to upgrade infer via robotized diminishing and fussing of the words into word roots. From index term by evacuating any connected prefixes and postfixes, Stemming is done to proceeding piece of work of index word, and more extensive idea than the real word is spoken by trunk. In an IR framework, the numeral of recovered archives is expanded by stemming process.**

***Keywords*—Stemming; truncating; statistical; NLP; IR; Lovins; Porters; Paice/Husk; Dawson; N-gram; HMM; YASS**

I. Introduction

In present days, indexing and search systems support the word stemming and are in twist chunk of Natural Language Processing (NLP) systems, Information Retrieval (IR) systems and Text Mining applications. The principal concept is to ameliorate recollect by lessening the words to their root's word [1]. Before the function of the index word, stemming is done and the concept of stem is broader than the actual term. The number of completed forms is enlarged through stemming operation in IR systems. Before any related algorithm is actually applied, the summary, categorization and text clustering is also needed as chunk of the pre operation.

Generally the data castoff to stock besides get search calls within IR refers to the identification lists identified as lead words. Work in the IR systems involves only recent Pakistani language interests. The evolution of such structures is constrained by the dearth of inaccessibility of language assets and utensils in these languages.

Endeavors are made to create more powerful explore drives for stemmer. Nearly all IR systems are used to lessen the structural alternatives of a word to its root [2]. Generally IR systems used to tokenize printed archives and use stemming to lessen the quantity of marks and catch semi-identical standings resulting from the root [3].

II. Problem Statement

Stemmer is unique method that uses the information retrieval system to lessen a word's structural alternatives to its stem. In recent years, the huge growth in the content of the Urdu and Sindhi web has enlarged the necessity for active algorithms and stemming methods. Stemmer allows us to lessen a word to its root. The execution of stemming algorithms in information retrieval has been an extended-ranking issue.

Now many algorithms for stemmers of different languages embracing language founded on Arabic Script have been designed and suggested. Yet small search is documented in the prose regarding the Sindhi and Urdu languages. The aim of study is to suggest generic stemmers, thorough explanations, deliberations and assumptions of the numerous procedures of recycled for abbreviating and arithmetical mechanisms, approaches and devices that are in trend and have been recycled and applied before.

III. Objectives

- To study numerous methods from the literature that were recycled and executed for the Stemming of the numerous languages
- To analysis the algorithmic mechanisms of automatic stemmers' Truncating and Statistical approaches.
- For stemmers of various languages, the tests of the methods used Truncating and the statistical algorithms are previously performed.
- To examine cause and issues behind intended consequences based on stated results of certain algorithms.
- To suggest a more suitable stemmer algorithm for Pakistani languages i.e. Urdu and Sindhi which could yield near outcomes satisfactory.

IV. Overview of Stemming

Stemming is the procedure of reducing the amount of modulated words to find root and is the support mechanism for numerous NLP applications with the IR method meanwhile the search process is based only on the word's stem. Stemming gives an IR system two significant advantages. First, it improves the system's recall as the query words are harmonized in the documents with their morphological versions and second it lessens catalogue scope resulting in noteworthy advantages in haste and retention necessities. Following Table I displays the stemming system.

TABLE. I. Stemming System

Prefix	Stem	Suffix	Complete Word	Meaning
نا	اميد	ي	نا اميدي	Hopelessness
اث	لک	يا	اثليکيا	Unwritten
وذ	قّر	و	وذقّرّو	Heavy Rain
پر	ديس	ي	پرديسي	Foreigner

V. Arrangement of Stemming Algorithms

It is possible to classify stemming algorithms into classes. There is a typical way for each of these groups to find the stems of word variants. Fig. 1 displays the arrangement of stemming algorithms.

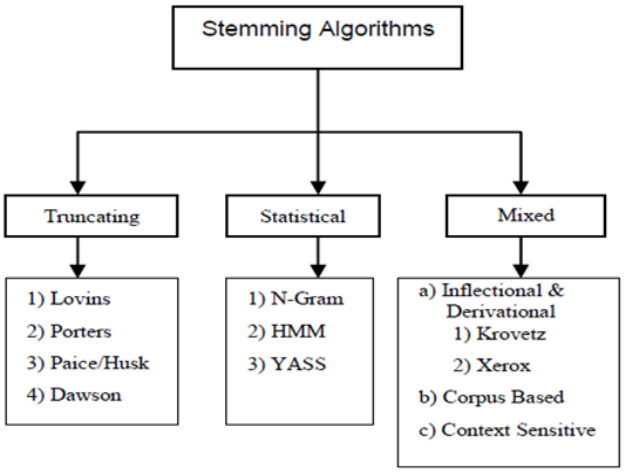

Fig. 1. Arrangement of Stemming Algorithms.

VI. Truncating Methods (Affix Removal)

The approaches are associated to eliminating a words' suffixes or prefixes (usually referred to an affixes), as clearly suggested by name. This was a simplest stemmer which shortened a term at the nth sign. Terms shorter than n are held as they are in this system. Over stemming chances increase when the length of the word is short.

Another modest method was the S-stemmer, a procedure that combines singular and plural noun shapes. Donna Harman suggested this algorithm [4].

The algorithms have rules for the deletion of plural suffixes so that they can be transformed to the unique forms. Truncating algorithms are the most widely used stemmer.

A. Lovins Stemmer

This was Lovins' initially prevalent and operative stemmer in 1968. It achieves a lookup on 294 end tables, 29 conditions and 35 rules for transformation. The stemmer of Lovins eliminates from a phrase the longest suffix. Upon removal of the ending, the term is recoded by a dissimilar table that requires different adjustments to translate these trunks into valid words. Unpaid to its flora as a single license procedure it always eliminates an extreme of single suffix from a word.

B. Porters Stemmer

Porters stemming algorithm was proposed in 1980 as the most popular stemming method. Many modifications and improvements on the basic algorithm were made and suggested [5]. It has five phases and directional are functional in each step until the criteria are passed by one of them. If a rule is adopted, the suffix will be deleted and the next step will be taken.

C. PAICE / HUSK Stemmer

This stemmer indexed to approximately 120 directions by the preceding memo of a suffix [6]. On each repetition, the past character of the term tries to search an appropriate law. Every law defines a termination, deletion or replacement. If no such law exists, it will stop.

D. Dawson Stemmer

It provides a large additional complete gradient of around 1200 suffixes. It has a one-pass stemmer too, so it's pretty quick. The suffixes are classified by their length and last letter in the reversed order indexed.

VII. Statistical Methods

Another solution to suffix striper is suggested by Prasenjit [7]. These stemmers depend on strategies and factual investigation. For example, most statistical stemmers used the Hidden Markov Model approach based on N-Gram. Melucci proposed a model using automatons of finite-state where the function of probability regulates transitions between states.

A. N-Gram Stemmer

An N-gram is a sequence of n characters, typically contiguous, extracted from a continuous text segment to be precise, a N-gram is a traditional of n consecutive characters dig out from a word. The key clue behind this approach is that a high proportion of N-grams will be shared by similar words.

B. HMM Stemmer

Melucci and Orio suggested this model [8]. In this method, it is probable to calculate the possibility of each track and invention the most likely path by the Viterbi coding in the automatic graph. To apply HMMs for stemming, the product of a concatenation of two subsequences can be viewed as a arrangement of letters that makes a word, a prefix and a suffix.

C. YASS Stemmer

The presentation of a stemmer produced by groups a wordlist without any dialect input is corresponding to that achieved using ordinary rule based stemmers like Porter's according to the authors. Groups are recognized using tiered approach and space measurements. The resulting clusters are

then measured to be classes of equivalence and their centroids to be the stems.

VIII. Literature Review

Since the decade, several stemmers have been produced and available on the computer and internet market. It is noted that stemmer is necessary part of any culture's process of gathering knowledge. Stemmer is the fundamental component of the IR process. Stemmers have been found to be the simplest type of all morphological systems. Since the absolute starting point of the information recovery period, the main group focus was established to support various languages and efficient algorithms. Majority of the stemmers work achieved in advance changes into based totally on regulations. The linguistic inputs based on the preparation of rules are very complex and hard work. In addition, it calls for excellent linguistic understanding to design such stemmers. Earlier stemmers were designed for the English language on a rule based approach. The first rule based stemmer was developed by Lovins [9]. Around 260 language rules have been mentioned for this purpose in order to curb the English language. Lovins' approach was the heuristic iterative longest match. Martin porter offered the most outstanding effort in the field of rule based stemmer [10]. He condensed Lovin's laws to roughly 60 guidelines. Porter stemmer algorithm, he has developed. This algorithm is very simple, effective and commonly used for search engine creation.

Urdu is a well-spoken language throughout the world and much work has been done on the stemming of Urdu. Riaz explained the Urdu stemming challenges and introduced a rule based model with a few rules that were enforced to inspire the specifics for Urdu [11]. This showed that originating from Urdu, due to the complex nature of Urdu is quite difficult.

Kansal has proposed a rule based on stemmer [12]. He established and implemented rules for this purpose to eliminate the suffix and prefix from the inflected words of Urdu. In addition the rule based stemmer for the Urdu language was created by Gupta [13].

By applying the truncation of affixes, light weight Stemming is to find a representative type of word indexing [14]. In Urdu, for a single word form there are large numbers of variant variants. Khan raised a number of morphological questions relating to Urdu stemmer's law-based development [15].

IX. Experiments and Results

A. Results Recorded using the Lovins Stemming AlgoritHM

Various writers castoff the Lovin's stemming algorithm to measure corpus accuracy by using various languages such as English, Urdu, Arabic and Sindhi. Table II shows the Lovin's stemming algorithm's precision.

With 99.1% and 93.37% precision, Wahiba and Sandeep used English, Haider used 100% precision of Arabic Language. When using 20583 words and 50000 characters, Rohit and Qurat-ul-ain used urdu with 85.15% and 91.2% precision. Fig. 2 demonstrates the precision of terms by giving different languages to readers.

TABLE. II. Accuracy of Lovin's Stemming Algorithm

Title	Author	Language	Corpus	Accuracy %
A new stemmer to improve Information Retrieval	Wahiba et al..	English	30000 Words	99.1
An effective stemmer in Devanagari script	Dogra et al.,	Devangari	1670 Words	94.26
Strength and accuracy analysis of Affix Removal Stemming Algorithms	Sandep et al.,	English	29417 Words	93.37
A Rule Based Extensible stemmer for information retrieval with application to Arabic	Haidar et at.,	Arabic	-----	100
Rule based Urdu Stemmer	Rohit et al.,	Urdu	20,583 Words	85.15
Assas-Band, an Affix-Exception-List Based Urdu Stemmer	Qurat-ul-Ain et al.,	Urdu	50,000 Words	91.2
Stemmer of Sindhi Secondary words for Information Retrieval System Using Rule based stripping Approach	Mohsin	Sindhi	50,327 Words	84.85

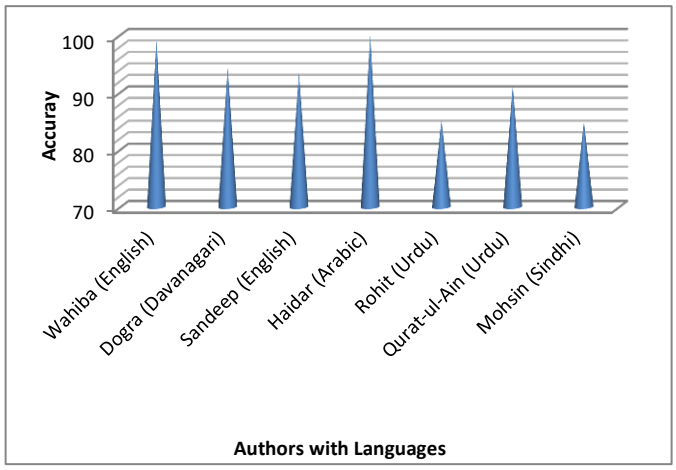

Fig. 2. Lovins Algorithm Stemming Accuracy.

B. Results Recorded using the Porters Stemming Algorithm

Various writers recycled this algorithm to measure the reliability of the corpus by using different languages. The accuracy of the Porters stemming algorithm is discussed in Table III.

Sandeep, Joshi and Wahiba used English with 73.61 percent, 98.4 percent and 99.5 percent accuracy. Widjaja used Indonesian with 96.31 percent accuracy, Guastad used Dutch with a consistency of 79.23 percent when using 45000 words. Fig. 3 demonstrates the accuracy of the concept when offering different languages to authors.

TABLE. III. PORTERS ACCURACY OF THE STEMMING ALGORITHM

Title	Authors	Language	Corpus	Accuracy%
Implementation of Porters Modified Stemming Algorithm in an Indonesian word Error Detection Plug in Application	Widjajja et al.,	Indonesian	3000 Words	96.31
A new stemmer to improve information Retrieval	Wahiba et al.,	English	30000 Words	99.5
Accurate stemming of Dutch for text classification	Gaustad et al.,	Dutch	45000 Words	79.23
Development of a stemmer for the Greek language	Georgios et al.,	Greek	880 Words	92.1
Strength and accuracy analysis of Affix removal stemming algorithms	Sandeep et al.,	English	29417 Words	73.61
Modified Porter Stemming Algorithm	Joshi et al,	English	30K Words	98.4

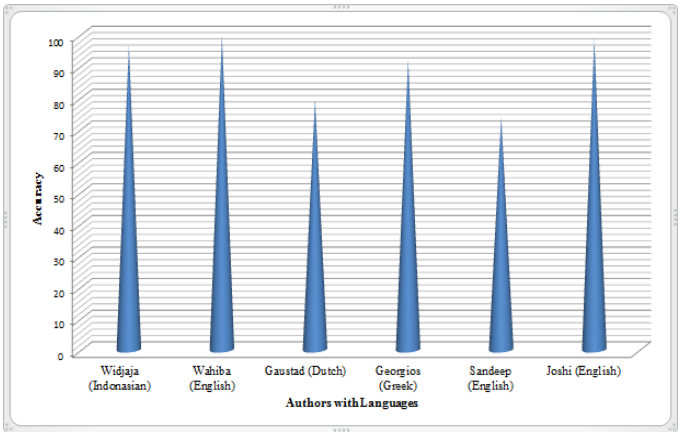

Fig. 3. Porters Precision Stemming Algorithm.

C. Results Recorded using the PAICE / HUSK Stemming Algorithm

Various authors used the Paice / Husk Stemming Algorithm to calculate corpus accuracy using various languages such as English and Portuguese. The accuracy of the Paice / Husk Stemming Algorithm is explained in Table IV.

Wahiba, Chris and Sandeep utilized English with 99.3 percent precision, 67 percent precision and 95.47 percent precision. Orengo utilized Portuguese dialect with 96 percent precision while the utilization of 30000 words. Fig. 4 appears phrase precision via providing readers exclusive languages.

D. Results recorded using N-Grams Stemming Algorithm

Various authors used the stemming algorithm N-Grams to calculate corpus accuracy using different languages such as Arabic, Malay, Marathi and English. The N-Gram stemming algorithm accuracy is explained in Table V.

TABLE. IV. PAICE / HUSK STEMMING ACCURACY ALGORITHM

Title	Authors	Language	Corpus	Accuracy%
Strength and Accuracy Analysis of Affix removal stemming algorithms	Sandeep et al.,	English	29417 Words	95.47
A new stemmer to improve information retrieval	Wahiba et al.,	English	30000 Words	99.3
Stemming algorithm for the Portuguese language	Orengo et al.,	Portuguese	2800 Words	96
An Evaluation Method for Stemming Algorithm	Chris et al.,	English	9,757 Words	67

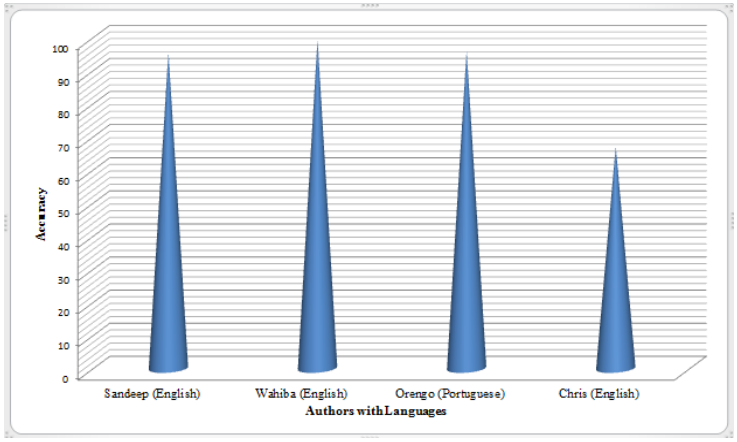

Fig. 4. Precision of PAICE / HUSK Stemming Algorithm.

TABLE. V. N-GRAMS PRECISION STEMMING ALGORITHM

Title	Authors	Language	Corpus	Accuracy %
Generation, implementation and Appraisal of an N-gram based Stemming Algorithm	Oande et al.,	English	COCA	96.7
Discovering suffixes: A case study for Marathi language	Majgao-nker et al.,	Marathi	1500 Words	82.50
Corpus-Based Arabic Stemming Using NGrams	Zitouni et al.,	Arabic	1000,000 Words	99.7
Effectiveness of stemming and Ngrams String Similarly Matching on Malay Documents	Sembok et al.,	Malay	2238 Words	98.2
Comparison of Stemming and N-Grams Matching for term Conflation in Arabic Text	Hani et al.,	Arabic	50000 Words	96

With 99.7 percent and 96 percent accuracy, Zitouni and Hani used Arabic. Pande used English language is 96.7 percent accuracy. Majgaonke recycled Marathi language when using 1500 words, which was 82.50 percent accuracy. Sambok used Malay language for 98.2 accuracy when using 2238 words. Fig. 5 demonstrates word accuracy by giving different languages to writers.

E. Results Recorded using HMM Stemming Algorithm

Various authors used the HMM Stemming Algorithm to calculate the accuracy of corpus by using different languages such as Arabic, Persian, Assamese and English. Table VI shows the HMM stemming algorithm accuracy.

Alajmi used 15 Million words to get 95 percent accuracy in English. Massimo used Arabic Language 90.5 percent were right when using the 1950 terms. Using 500 letters, the Persian language used by Fatimah was 79 percent correct. Navanath used Assamese language, 92 percent accuracy when using 2000 words. Fig. 6 displays the accuracy of the word by giving writers different languages.

F. Results Recorded using YASS Stemming Algorithm

Several researchers used different titles to calculate corpus accuracy when using the YASS stemming algorithm with different languages like English, Hungarian and Kebang. The precision of this stemming algorithm is shown in Table VII.

Prasenjit using English with the precision of 96.5 percent when using 262128 letters. Prasenjit also had 86.68 percent accuracy when using 536678 letters.By using 30000 letters, the accuracy of Sadiq used Kebang language was 87 percent. Fig. 7 demonstrates word accuracy by giving different languages to readers.

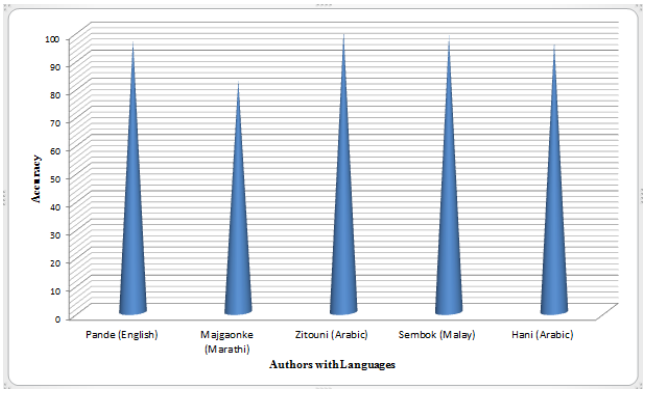

Fig. 5. N-Gram Precision of the Stemming Algorithm.

TABLE. VI. HMM ALGORITHM STEMMING ACCURACY

Title	Authors	Language	Corpus	Accuracy%
An improved stemming approach using HMM for a highly inflectional language	Navanath et al.,	Assamese	2000 Words	92
PHMM: Stemming on Persian Texts using Statistical Stemmer based on hidden Markov Model	Fatemeh et al.,	Persian	500 Words	79
Hidden markov model based Arabic morphological analyzer	Alajmi et al.,	English	15 Million Words	95
A novel method for stemmer generation based on hidden markov models	Massimo et al.,	Arabic	1950 Words	90.5

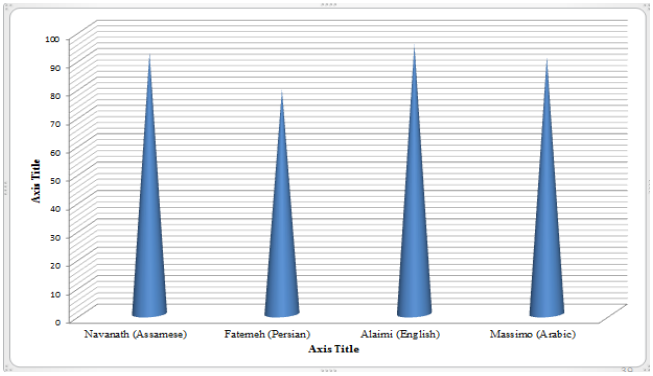

Fig. 6. Precision of HMM Stemming Algorithm.

TABLE. VII. YASS ALGORITHM STEMMING ACCURACY

Title	Authors	Language	Corpus	Accuracy%
YASS: Yet Another Suffix Stripper	Prasenjit et al.,	English	262128 Words	96.5
Hungarian and Czech Stemming using YASS	Prasenjit et al.,	Hungarian	536678 Words	86.68
The First Step Towards Suffix Stripping of Missing Words using YASS	Sadiq et al.,	Kebang	30,000 Words	87

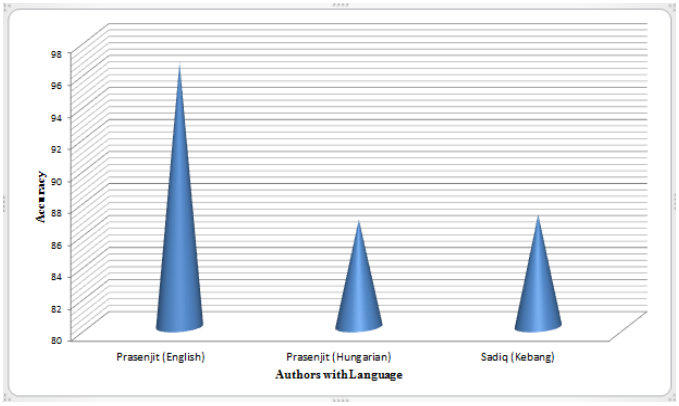

Fig. 7. YASS Precision Stemming Algorithm.

X. Discussion

The steps of Dawson Stemming Algorithm are similar as Lovins algorithm in a great extent. Hence, researchers use Lovins algorithm in its place of Dawson Algorithm.

By using the Lovins stemming algorithm, 100% accuracy is calculated by Haidar using the corpus of Arabic language. And maximum accuracy is reported with the data set of English which is 99.1%. This kind of algorithm is also used with Sindhi by Mohsin but he has not achieved good results as relate to Arabic and English since partial linguistics rules were applied. If he raises the commands then accuracy may also increase.

Wahiba and Sandeep implemented Lovins, Porters and Paice/ Husk algorithms on the corpus of English and accomplished acceptable level results.

Among the statistical stemming algorithms, most of the researchers used N-Grams Based algorithm for the task of stemming. Zitouni achieved 99.7% accuracy with Arabic and Sembok calculated 98.2% accuracy with malay. Concluded the nature of both languages are entirely opposite from each other but due to the N-gram language modeling is does not affect the performers of the stemmers.

Limited number of researchers used YASS stemming algorithm for researchers believed that this algorithm is tough in terms of implementation and require more time for execution as compare to other statistical stemming algorithms.

XI. Conclusion

A relative evaluation of statistical and truncating algorithms in specific languages with in the literature is provided. The work purposes to suggest for the dissimilar dialects a generic stemming annotation. Truncating algorithms, especially Porters and Lovins, have been observed to be more appropriate for Sindhi, Urdu and other languages based on Arabic scripts since both processes are working on the specific rules of linguistics.

XII. Future Work

Although much research has already been done in the development of stemmers, much remains to be done to advance the accuracy.

A stemmer that uses both syntactic and semantic knowledge to reduce stemming errors should be developed.

For the Sindhi stemming method, the Porters and Lovins algorithms could be used to evaluate which set of rules is more appropriate for Sindhi.

Acknowledgment

This research has been conducted in SALU Khairpur as part of Master's dissertation.

References

[1] Jivane, A. G., (2011), "Comparative Study of Stemming Algorithms", International Journal of Computer Technology Applications, Vol. 2, Number 6, Pp. 1930-193.

[2] Husain, M. S., (2012), "An Unsupervised Approach To Develop Stemmer", International Journal on Natural Language Computing, Vol. 1, Number.2, Pp. 1523.

[3] Al-Omari, A., Abuata, B., (2013), "Building and Benchmarking New Heavy/Light Arabic Stemmer", The 4th International Conference on Information and Communication Systems.

[4] Harman Donna, (1991), "How effective is suffixing?" Journal of the American Society for Information Science, Vol. 42, Pp. 7-15 7.

[5] Lovins, J. B, (1968), "Development of a Stemming Algorithm," Mechanical Translation and Computer Linguistic, Vol.11, Number 1/2, Pp. 22-31.

[6] Porter M.F, (1980), "An Algorithm for Suffix Stripping", Program, Vol. 14, Pp. 130-137.

[7] Prasenjit, M., Mandar, M., Swapan K. Parui, G., Pabitra, M., (2007), "YASS: Yet another Suffix Stripper", ACM Transactions on Information Systems, Vol. 25(4).

[8] Melucci, M., Orio, N., (2013), "A Novel Method for Stemmer Generation based on Hidden Markov Models", Proceedings of the Twelfth International Conference on Information and Knowledge Management, Pp. 131-138.

[9] Lovi7*+ns, J. B., (1968), "Development of a stemming algorithm", Mechanical Translation and Computational Linguistics, Volume 11, Pp. 22–31.

[10] Porter, M., (1980), "An Algorithm for Suffix Stripping". Program, Volume 14, Number 3, Pp. 130-137.

[11] Riaz, K., (2007), "Challenges in Urdu Stemming", A Progressive Report In BCS IRSG Symposium: Future Directions in Information Access (FDIA 2007). http://citeseerx.ist.psu.edu/viewdoc/download/ 1 0.1.1.102.3051.pdf

[12] Kansal, R., Goyal, V., Lehal, G. S., (2012), "Rule Based Urdu Stemmer", Proceedings of COLING 2012, Pp. 267–276.

[13] Gupta, V., Joshi, N., Mathur, I., (2013), "Rule Based Stemmer in Urdu", In Proceedings of 4th International Conference on Computer and Communication Technology, Pp. 129-132.

[14] Al-Sughaiyer, I., Al-Kharashi, I., (2004), "Arabic morphological analysis techniques: a comprehensive survey", Journal of the American Society for Information Science and Technology, Volume 55, Number 3, Pp. 189 – 213.

[15] Khan, S. A., Anwar, W., Bajwa, U. I., Wang, X., (2011), "Challenges in Developing a Rule based Urdu Stemmer", Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing, Pp. 46–51.