Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 11, November 2014, pg.1 – 9

RESEARCH ARTICLE

A Hybrid Approach used to Stem Punjabi Words

Puneet Thapar[*]
CSE Department, DAVIET College, Jalandhar
puneet.thapar90@gmail.com

Abstract— Stemming is the process of removing the affixes from inflected words, without doing complete morphological analysis. A stemming Algorithm is a procedure to reduce all words with the same stem to a common form [20]. The purpose of stemming is to obtain the stem or radix of those words which are not found in dictionary. If stemmed word is present in dictionary, then that is a genuine word, otherwise it may be proper name or some invalid word. It is useful in many areas of computational linguistics and information-retrieval work. This technique is used by the various search engines to find the best solution for a problem. The algorithm is a basic building block for the stemmer. Stemmer is basically used in information retrieval system to improve the performance .The paper present a stemmer for Punjabi, which uses a Naive algorithm. We also use a suffix stripping technique in our paper. Similar techniques can be used to make stemmer for other languages such as Hindi, Bengali and Marathi. An in depth analysis of Punjabi news corpus was made and various possible noun suffixes were identified like ੀ ਆਂ īāṃ, ਿੀਆਂ iāṃ, ੀ ਆਂ ūāṃ, ੀ ਾਂ āṃ, ੀ ਏ īē etc. and the various rules for noun and proper name stemming have been generated. The result of stemmer is good and it can be effective in information retrieval system. This stemmer also reduces the problem of over-stemming and under-stemming.

Keywords— Stemmer, Stemming, Naive Algorithm, Suffix Striping, Under-stemming, Over-stemming, Stemming Algorithm

I. INTRODUCTION

Stemming is well known technique in information retrieval system to improve the ability to match query and document vocabulary. Stemming is used to find the root word from an inflected word. It cut the inflection from the word and we will get the root word. Stemming is used in various languages. All the languages like English, Hindi, Marathi, Nepali, Bengali uses stemming in their information retrieval systems. The stem need not be identical to the morphological root of the word, it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. A stemmer for English, for example, should identify the string cats and possibly catlike, catty etc. as based on the root cat, and stemmer, stemming, stemmed as based on stem. A stemming algorithm reduces the words fishing, fished, fish, and fisher to the root word, fish. Stemming is an operation that conflates morphologically similar terms into a single term without doing complete morphological analysis. Stemming (Haidar et al., 2006) is used in information retrieval systems to improve performance. Stemming is done to improve the efficiency of various search engines to get the result. Stemming has also been applied to text classifications. Stemming is an attempt to make the Search engines more effective in information retrieval. For grammatical reasons, documents are going to use different forms of a word, such as table, tables. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. Stemming makes group of derivationally related words with similar meanings represented single meaning.

The first paper on the stemmer was published in 1968.It was written by Julie Beth Lovins [1]. A later stemmer was written by Martin Porter and was published in the July 1980 issue of the journal Program [2]. E.g. table 1 shows the stemming examples.

Table 1: Stemming Example

INFLECTED WORD	ROOT WORD
ਕੁਤੇ	ਕੁਤਾ
ਮੁੰਡੇ	ਮੁੰਡਾ
ਕਮਰੇ	ਕਮਰਾ
ਜਮਾਤਾ	ਜਮਾਤ

In Punjabi language stemming (Mandeep et al.,2009) for nouns and proper names, an attempt is made to obtain stem or radix of a Punjabi word and then stem or radix is checked against Punjabi noun morph and proper names list. An in depth analysis of Punjabi news corpus was made and various possible noun suffixes were identified like ◌ੀ ਆਂ īāṃ, ਿ◌ੀਆਂ iāṃ, ◌ੀ ਆਂ ūāṃ, ◌ੀ ◌ੀਂ āṃ, ◌ੀ ਏ īē etc. and the various rules for noun and proper name stemming have been generated. Punjabi language stemmer for nouns and proper names is applied for Punjabi Text Summarization. Text Summarization is the process of condensing the source text into shorter version. Those sentences containing Punjabi language nouns or proper names are important.

II. BACKGROUND AND RELATED WORK

Research in Punjabi Stemming is not done so far as compared to the other languages. The stemmer for other languages like English, Nepali, Bengali and Hindi are present. Mostly the word is done on English language. Algorithm for suffix stripping is used in 1980 by M.F Porter. In this it uses a list of suffixes by which it matches an inflected word and removes the suffix [2]. Stemming algorithm is used for German languages. In this stemmer firstly it removes a suffix from the word and then checks the validity of word. If the word found to be illogical then it substitutes the suffix with the other words [4]. In the Dutch stemmer it uses a suffix stripping algorithm and dictionary lookup rule based methods [5].In the Nepali Stemming it uses a morphological analyzer which determines the given inflected word .In this it also tells about the Dawson stemming algorithm, krowertz algorithm [6].Lightweight Stemmer for Bengali also exists. In which it just strips the affix from the word without doing the complete morphological analysis. It removes suffixes as well as prefixes. This type of approach that is used for stemming is also called affix removal approach [7]. In the lightweight stemmer for Hindi it uses a look up table approach in which word is matched with the words present in the lookup table. Light weight stemmer approach uses affix removal algorithm and n gram stemming algorithm. It also shows the over stemming errors and the under stemming errors [13].There is a hybrid approach which is used for stemming of Arabic text. In this approach it uses a dictionary technique, morphological analysis, affix removal, statistical and translation technique. It also shows the accuracy of this hybrid approach on various areas like economics, science, medical and sport [14].

III.PUNJABI LANGUAGE DESCRIPTION

Punjabi is called also Gurmukhi or Shahmukhi. It was developed by Guru Nanak Dev ji (first Sikh Guru) in the 16th Century. Gurmukhi means "from the mouth of the Guru"[20]. Shahmukhi is mostly spoken in Pakistan and written in Arabic text. Arabi has a different syntax than the Punjabi text. Shahmukhi is not so much spoken in Punjab. Shahmukhi have different writing syntax than the Gurmukhi Language. Due to writing syntax in Arabic this is most popular in Pakistan. Gurmukhi has its own features like it is a tonal language with three tones. In this when the consonants are used together, then special symbols are used together to combine the rest part of the word. Here is the list of vowels, laga matra, consonants and the other symbols that are used in Gurmukhi. By using these components you can learn and speak Punjabi easily. In the section 6, it contains the list of symbols and in the brackets these is a text written that means how to pronounce the Punjabi characters. Guru Granth Sahib is written in Gurmukhi script. Modern Gurmukhi has 41 consonants and 9 vowel sounds

Table 2: Laga matra's in Punjabi language

S.no.	Symbols	Names
1.	ਿ	Sihari
2.	ੀ	Bihari
3.	ਾ	Kanna
4.	ੁ	Aonkar
5.	ੂ	Dulonkar
6.	ੇ	Lanvan
7.	ੈ	Dulanvan
8.	ੋ	Hora
9.	ੌ	Kanora

Other symbols in Punjabi

ੱ ਅਦਕ(adhak)	ਂ ਬਿੰਦੀ(bindi)	ੰ ਟਿੱਪੀ(tippi)

Consonants in Punjabi language

ੳ (ura)	ਅ (aira)	ੲ (iri)	ਸ (sassa)	ਹ (haha)
ਕ (kakka)	ਖ (khakha)	ਗ (gaga)	ਘ (ghaga)	ਙ (nanna)
ਚ (chacha)	ਛ (shasha)	ਜ (jaja)	ਝ (jhaja)	ਞ (naimna)
ਟ (taimka)	ਠ (thatha)	ਡ (dadda)	ਢ (dhadda)	ਣ (naanna)
ਤ (tatta)	ਥ (thattha)	ਦ (dadda)	ਧ (dhadhha)	ਨ (nana)
ਪ (pappa)	ਫ (fafa)	ਬ (baba)	ਭ (bhabha)	ਮ (mamma)
ਯ (yaya)	ਰ (rara)	ਲ (lala)	ਵ (vavva)	ੜ (rarha)

ਸ਼ sassha	ਖ਼ khakhha	ਗ਼ gaggha	ਜ਼ jajjha	ਫ਼ faffha	ਲ਼ lallha

IV. AVAILABLE STEMMING TECHNIQUES

Many theories and experiments have been developed to evaluate the efficiency and the stability of the stemming process in information retrieval. There are several techniques used for word stemming, developed within the time. The first approach was Dictionary-Based approach then Corpus-Based Technique and then the rule based techniques.

A. Dictionary-Based Technique

Historically, stemmers have often been thought of as either dictionary-based or algorithmic. In the Dictionary based stemmers it matches every word with a word on a proper digitalized dictionary, correspond each word to its stem. This method seems to be effective but inadequate to deal with unlimited words and their formation. This was the main reason that led him to evaluate algorithmic stemmers and conclude that “despite the errors they can be seen to make, they still give good practical results”. Dictionary-based stemmers require dictionary maintenance, to keep up with an ever-changing language, and this is actually quite a problem. It is not only that a dictionary created to assist stemming nowadays will probably require major updating in a few years’ time, but also that a dictionary in use for this purpose today. [15]

B. Rule-Based Technique

This is the widest applied stemming technique, with most representative the algorithm introduced by Porter (1980). With specific rules for the English language, this algorithm removes iteratively suffixes from a given word, reducing it on its stem. Even if the algorithm has its limitations, it is the most commonly accepted for its high precision and recall. Lovin’s stemmer (1968) follows the same rule-based technique but it does not apply its rules iteratively and it is more conservative than Porter’s algorithm. On this path Paice & Husk (1990) have also worked introducing one more English stemmer with different rules. For Scandinavian languages we have also rule-based stemmers presented on 2001. Finally, applying the same philosophy, there are implemented rulebased stemmers for Romance languages (English, French, Spanish, Portuguese, Italian), Germanic Languages (German, Dutch), Scandinavian Languages (Swedish, Norwegian, Danish), Other Languages like Russian [15].

C. Light-Stemming Technique

Today there are plenty of rule-based algorithms and stemmers, developed for various languages. Most of the times, for each of them, a different algorithm is used to reach higher precision in the results. So lately we have light-stemmers, referred to the process of stripping off a small set of prefixes and/or suffixes without trying to deal with infixes or recognize patterns and find roots. [15]

D. Corpus-Based Technique

An algorithmic stemmer uses lists of words either for suffix removal or exclusion. The more advanced is the algorithm the longer are these lists. On the other hand, a dictionary-based stemmer needs to remove some basic suffixes before starts the look-up process in the extended dictionary. Trying to improve the effectiveness of these stemmers we are driven to the rule-based technique. This hybrid perspective was applied in many stemming algorithms. The hypothesis of that work is that the word forms that should be conflated will co-occur in documents from the corpus. Corpus-based stemming was found to provide moderate improvement over existing rule-based stemmers. [15]

E. Lovins Stemmer

The lovin stemmer is a single pass stemmer, context sensitive and longest match stemmer. Lovins stemmer maintains a list of most frequent suffixes, 250 in number and removes the longest suffix ensuring that the stem is at least three characters long. It is unreliable and frequently fails to produce the correct stem. [6]

F. Porter Stemmer

This is the conflation stemmer. This was designed for English language. This is made up of a combination of smaller and simpler suffixes. It has 5 steps applying rules with each step. If any suffix rule matches the word, then the condition attach to it are tested and stem is obtained by removing the suffix. It is mostly used stemmer. [6]

G. Dawson Stemmer

This stemmer is based on the Lovins stemmer. It extends the suffix list to 1200 suffixes. It keeps the longest match and single pass of Lovins stemmer.

H. Krovertz Stemmer

This stemmer is basically based on the morphology. It removes the suffixes and then checks the words in the dictionary. [6]

V. PROPOSED METHODOLOGY FOR STEMMING IN PUNJABI

Proposed stemmer is based on hybrid approach of suffix stripping algorithm and suffix substitution algorithm and some extent of Naive technique.

A. Naive Algorithm

The term Naive is the concept of artificial intelligence research and problem solving concept in mathematics denoted as Naive search. Naive stemmers employ a lookup table which contains relations between root forms and inflected forms. To stem a word, the table is queried to find a matching inflection. If a matching inflection is found, the associated root form is returned. Naive approaches are criticized for their general lack of elegance in that no algorithm is applied that would more quickly converge on a solution. In other words, there are more operations performed during the search than should be necessary. The algorithm is only accurate to the extent that the inflected form already exists in the database. Given the number of words in a given language, like English, it is unrealistic to expect that all word forms can be captured and manually recorded by human action alone. Manual training of the algorithm is overly time-intensive and the ratio between the effort and the increase in accuracy is marginal at best. Naive algorithms do overcome some of the challenges faced by the other approaches. Naive algorithms are initially very difficult to design given the immense amount of relations that must be initially stored to produce an acceptable level of accuracy. However, Naive algorithms are easy to improve in that decreasing the stemming error is only a matter of adding more relations to the table. Someone with only a minor experience in linguistics is capable of improving the algorithm, unlike the suffix stripping approaches which require a good knowledge. This term is also same as pattern matching algorithm in which word is matched word by word. The result depends on the matching of word.

B. Suffix Stripping Algorithms

Suffix stripping algorithms do not rely on a lookup table that consists of inflected forms and root form relations. Instead, a typically smaller list of "rules" is stored which provide a path for the algorithm, given an input word form, to find its root form. Some examples of the rules include:

- If the word ends in ' w ', remove the ' w '.
- If the word ends in ' W ', remove the ' W '.
- If the word ends in 'IAw', remove the ' IAw '.

Suffix stripping approaches enjoy the benefit of being much simpler to maintain than Naive algorithms, assuming the maintainer is sufficiently knowledgeable in the challenges of linguistics and morphology and encoding suffix stripping rules. Suffix stripping algorithms are sometimes regarded as crude given the poor performance when dealing with exceptional relations (like 'dosq' and 'dosqI').

VI. WORKING OF PUNJABI STEMMER

Firstly the word entered is checked in database of root words, here we are using Naive technique but only to check whether the entered word is root word or not. If it is found then it will be simply displayed as output, which means the word entered by the user is root word. If it doesn't exist in root word database then suffix stripping along with suffix substitution will be applied

In the proposed stemmer there will be two database table one table will be of root words having approximately 250000 root words and the other table will maintain the suffix to be stripped and corresponding suffixes to be substituted. So suffix stripping and suffix substitution will be done from the database of suffixes. After that the stemmed word will again be checked in the root word database table to ensure that the word is stripped correctly and will be displayed to the user. The proposed stemmer reduces the problem of over-stemming and under-stemming since the result is displayed to user after ensuring the stemmed word in the root word database i.e. if the stemmed word is over-stemmed or under-stemmed it will not match in the root word database. So, accurate results will be displayed to user.

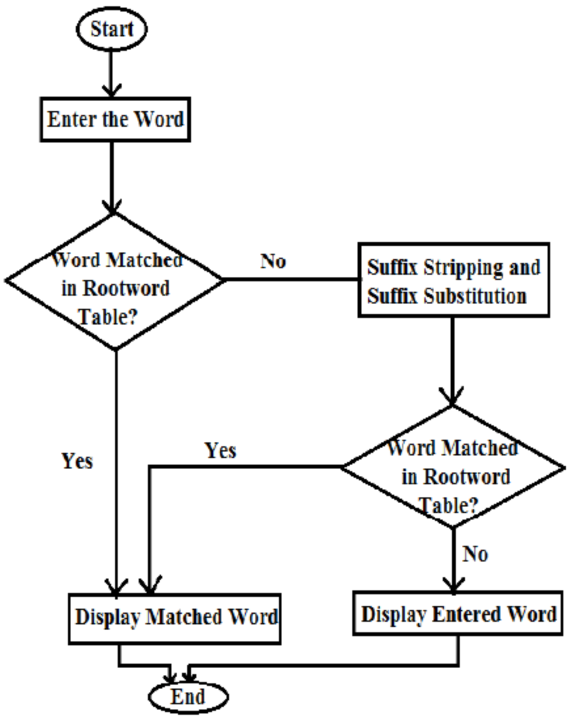

Start
Enter the Word
Word Matched in Rootword Table?
No
Suffix Stripping and Suffix Substitution
Yes
Yes
Word Matched in Rootword Table?
No
Display Matched Word
Display Entered Word
End

Figure1. Working of Punjabi Stemmer.

VII. EXPERIMENTAL RESULTS

A. *Correctness*

Correctness of a stemmer depends on the word present in the look up table. We have entered 52000 words in our lookup table therefore it reduces the chances of going to the second option of suffix stripping. Bigger database causes more correctness.

B. *Effectiveness*

Effectiveness of the stemmer depends on the behavior of the system. Behavior means what stemmer will do whenever an abnormal condition occur. Abnormal condition means whenever somebody tries to enter a word which does not exists. Our stemmer is very effective if anybody tries to enter a word that does not exists it just shows that word in the output box. Our stemmer effectively adjusts in different type of environment. Over stemming and under-stemming problem are also controlled in the stemmer. These errors are more whenever we are using suffix removal technique. By using Naive technique we have controlled these errors. If the word found in the database then there is no chance of these errors to occur.

C. *Performance*

Performance of stemmer will be high if the result is positive. The conflation class of stemmer is 4.Here in the above calculation it shows that we have 52000 words entered in the list and 14400 words are root type words we can calculate mean number of words by dividing the number of unique words with the number of unique stem after stemming.

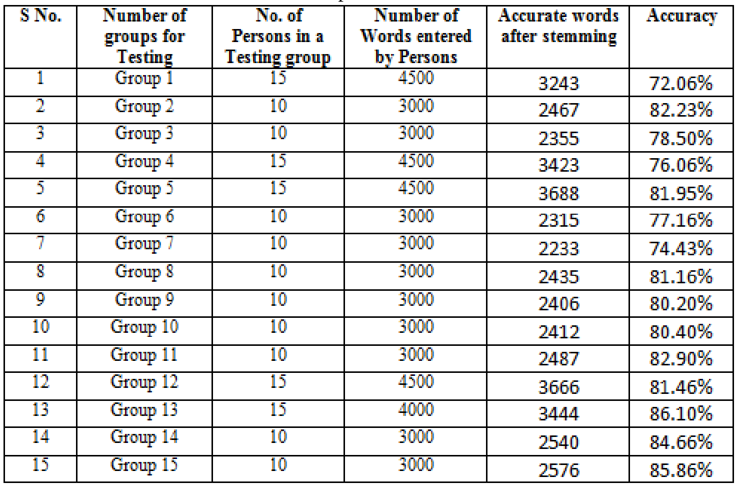
MWC=Mean Number of Words.
WC is given by the following equation.
MWC = N/S --------------- (1)
N=Number of unique words before stemming
S=Number of unique stem after stemming

In our stemmer number of unique word before stemming are 52000 and the number of unique words after stemming are 12700. By substituting these values in equation no 1 we get

MWC=52000/12700
MWC=4.09

We have tested our stemmer with the help of ten groups of persons. Results are shown in Table 3.
Here we have taken fifteen groups. Each group has different number of persons. Each group has different number of words to be tested. We have calculated the accuracy for each group and then calculate the accuracy of our stemmer by calculating the average accuracy of the groups. Average accuracy of our stemmer is 81.27%.

Table 3 Experimental Results



S No.	Number of groups for Testing	No. of Persons in a Testing group	Number of Words entered by Persons	Accurate words after stemming	Accuracy
1	Group 1	15	4500	3243	72.06%
2	Group 2	10	3000	2467	82.23%
3	Group 3	10	3000	2355	78.50%
4	Group 4	15	4500	3423	76.06%
5	Group 5	15	4500	3688	81.95%
6	Group 6	10	3000	2315	77.16%
7	Group 7	10	3000	2233	74.43%
8	Group 8	10	3000	2435	81.16%
9	Group 9	10	3000	2406	80.20%
10	Group 10	10	3000	2412	80.40%
11	Group 11	10	3000	2487	82.90%
12	Group 12	15	4500	3666	81.46%
13	Group 13	15	4000	3444	86.10%
14	Group 14	10	3000	2540	84.66%
15	Group 15	10	3000	2576	85.86%

D. *Mean Removal Rate*

Mean Removal Rate is measured by calculating the number of suffixes attached with any word. E.g. if a word in Punjabi like muMfw *is used the various variations for* muMfw *is* muMfy, muMfIAw. muMfIAw *have three suffixes at end,* muMfy *have one and* muMfw *itself has no suffix therefore total number of suffixes are 3.*

Table 4 Suffixes present with word

WORD	SUFFIX	NO. OF SUFFIXES
ਮੁੰਡਾ	-	0
ਮੁੰਡੇ	ੇ	1
ਮੁੰਡਿਆ	ਿਆ	3

Mean Removal Rate=Sum of all the suffixes/Number of suffixes
Mean Removal rate for word ask is=0+1+3/3 =1.33

E. Over-stemming and under-stemming

Over-Stemming and Under-Stemming are the two error measurements in stemming algorithms. Over-stemming is an error where two separate inflected words are stemmed to the same root. Over-stemming occurs when words that are not morphological variants are conflated. Under-stemming is an error where two separate inflected words should be stemmed to the same root. Under-stemming occurs when words that are indeed morphological variants are not conflated. We have created a list of Punjabi words in which these over and under stemming error come if we are using a suffix stripping approach. The list of words contains approximately 200 words. We are minimizing these errors by entering these 200 words with a best solution in the database. We have shown the list of words in Table 5.

Table 5 List of some words
(Over-stemming and Under-stemming errors)

ਦੋਸਤ	ਦੋਸਤਾਨਾ	ਅਮਲ	ਅਮਲੀ	ਅਦਾ	ਅਗਾਇਗੀ
ਮਸਤ	ਮਸਤਾਨਾ	ਅਮੀਰ	ਅਮੀਰੀ	ਅਦਾਜ	ਅਦਾਜਾ
ਜਾਗ	ਜਾਗਨਾ	ਸੰਗਤਰਾ	ਸੰਗਤਰੀ	ਰੰਗ	ਰੰਗੀਲਾ
ਅਚਾਰ	ਅਚਾਰੀਆ	ਲਾਲ	ਲਾਲੀ	ਅਧਿਅਕਸ਼	ਅਧਿਅਕਸ਼ਤਾ
ਦੀਵਾ	ਦੀਵਾਨਾ	ਡਕੈਤ	ਡਕੈਤੀ	ਅਧਿਕ	ਅਧਿਕਾਰ
ਮਰ	ਮਰਾਨਾ	ਡਾਕਟਰ	ਡਾਕਟਰੀ	ਅਧੀਨ	ਅਧੀਨਤਾ
ਯਾਰ	ਯਾਰਾਨਾ	ਡਾਟ	ਡਾਟੀ	ਅਨਉਚਿਤ	ਅਨਉਚਿਤਤਾ
ਜਾਰ	ਜਾਰਾਨਾ	ਢੋਲ	ਢੋਲੀ	ਅਨੰਤ	ਅਨੰਤਤਾ
ਚਿੰਤਾ	ਚਿੰਤਕ	ਢੋਗ	ਢੋਗੀ	ਅਨਪੜ	ਅਨਪੜਤਾ
ਜਾਗ	ਜਗਰਾਤਾ	ਤਸਕਰ	ਤਸਕਰੀ	ਅਨਰਥ	ਅਨਰਥਕ
ਅਜਾਦ	ਅਜਾਦੀ	ਤਕਨੀਕ	ਤਕਨੀਕੀ	ਹਨੇਰਾ	ਹਨੇਰੀ
ਅੰਜੀਰ	ਅੰਜੀਰਨ	ਤਬਾਹ	ਤਬਾਹੀ	ਅਨਾਚਾਰ	ਅਨਾਚਾਰਤਾ
ਅਜੀਵ	ਅਜੀਵਕਾ	ਤਰਾਸ	ਤਰਾਸੀ	ਨੀਦ	ਨੀਦਰਾ
ਜੁੜਨਾ	ਜੁੜਬਾ	ਤਲਾਸ਼	ਤਲਾਸ਼ੀ	ਅਨੁਦਾਨ	ਅਨੁਦਾਨੀ
ਅੰਤਰੰਗ	ਅੰਤਰੰਗਤ	ਤਵਾ	ਤਵੀ	ਮਾਨ	ਮਾਨਤਾ
ਪੀਲਾ	ਪੀਲੀਆ	ਮੋਜ	ਮੋਜਾ	ਗਿਆਨ	ਗਿਆਨੀ

VIII. COMPARISON WITH OTHER STEMMERS

Most important factor for measuring the effectiveness or comparison of any NLP base application is the accuracy. Table 6 shows the comparison of our proposed stemmer for Punjabi with the stemmers of other languages on the basis of accuracy. These stemmer's works on different languages and different number of words. These stemmers uses different algorithm for stemming.

Table 6 Stemmer Comparison

Language	Technique Used	No. of Words	Accuracy in %
Dutch	Dutch Porter Stemmer	45000	79.23
Arabic	Language Dependent Rules	Rule Based	75
Hindi	Suffix Removal Technique.	35977	88
Arabic	Morphological Analysis, Affix-removal and Dictionaries	10000	83
AfamOrono	Affix Removal by Subsituting rules	5000	94.84
Telugo	Corpus Based Statistical Approach	500	70.8
Malayalam	Finite state automaton rules	3000	90.5
Greek	Porter Algorithm	703	92.7
Punjabi	Brute Force Technique	52000	81.27

We can measure accuracy very well whenever we are working on the database of words. When we are using rule base strategies and suffix stripping approaches then we will measure the accuracy in terms of over stemming and under-stemming errors. Suffix stripping approaches also gives accuracy whenever we are entering a countable amount of words.

IX. CONCLUSION AND FUTURE SCOPE

Our stemmer works for Punjabi and this is a simplified version of stemmer. Naive technique requires no preprocessing of text before stemming a word. It basically emphasizes on Naive techniques. Naive technique emphasis on the word to be found in the database. There is a very little influence of suffix stripping algorithm in our stemmer. There is a problem of over-stemming and under-stemming comes under suffix stripping approach. We can also do suffix substitution with suffix stripping to avoid the problem of over stemming and under-stemming. We have tested our system by creating groups of person. In which we have created 10 to 15 people in a group. Each group has different number of words to test. They tested their words and each group have calculates different accuracy. Here the accuracy depends on the accurate words to be stemmed. Some groups find out the simple words and other finds out the complex words to stem, therefore their accuracy differs. Our recommendation about this is that we have created a big database. It consumes lot of time. Accuracy depends on the size of the database. More words more accuracy. Big database also requires huge amount of storage capacity. Future scope of our research work is that we can create stemmer for Punjabi by using some other stemming algorithm. We can also increase the accuracy of stemmer by maximizing size of database. We can also compare the accuracy of our stemmer with other stemmers.

REFERENCES

[1] David A Hull Gregory Grefenstette (1996)" A Detailed Analysis of English Stemming Algorithms "Rank Xerox research Centre 6 chemin day mauperutis, 38240 Melyanfrance,pp 1-16.

[2] Julie Beth Lovins, (1968)"Development of a Stemming Algorithm*"Mechanical Translation and Computational Linguistics, Vol No.11, Issue No.1, pp 22-31.

[3] Ananthakrishnan Ramanathan and Durgesh D Rao(2003) "A Lightweight Stemmer for Hindi" In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL), on Computatinal Linguistics for South Asian Languages (Budapest, Apr.) Workshop,pp 42-48.

[4] Debela Tesfaye and Ermias Abebe (2010) " Designing a Rule Based Stemmer for Afaan Oromo Text" Iinternational Journal of Computer Linguistics, Vol. No.1, Issue No.2 , pp 1-11.

[5] Tanja Gaustad and Goose Bauma (2000)"Accurate Stemming of Dutch for Text Classification" Language Computing. Vol No.45, Issue No. 1, pp 104-117.

[6] M. Santosh Kumar and Kavi Narayana (2006)"Corpus Based Statistical approaches for stemming telugo" Journal of quantative linguistic, Vol. No.16, Issue No.1 ,pp 130-133.

[7] Tuomo Korenius, Jorma Laurikkala, Kalervo Järvelin, Martti Juhola (2009) "Stemming and Lemmatization in the Clustering of Finnish Text Documents"Proceedings of 13th ACM international conference on information and knowledge management Vol. No. 45,Issue No. 6, pp 714-723.

[8] Eiman Tamah Al-Shammari (2008)"KTOWARDS AN ERROR-FREE STEMMING" URL: www.iadis.net/dl/final_uploads/200812R027.pdf, pp 160-163.

[9] Payas Gupta (2006)"Stemmer for Indo-Aryan Languages: A survey" URL: www.mysmu.edu/phdis2008/payas.gupta.2008/papers/IS701.pdf.

[10] Lily Suryana Indradjaja and Stephane Bressan (2003)"Automatic Learning of Stemming Rules for the Indonesian Language"In the Proceedings of the 17th Pacific Asia Conference, pp 62-68.

[11] Vijay Sundar Ram R, Pattabhi R K Rao T and Sobha Lalitha Devi(2010) "Malayalam Stemmer "in Computational Linguistics Research Group Vol No. 14, issue No. 3, pp 130-137.

[12] Mudassar M. Majgaonker and Tanveer J Siddiqui (2010) "Discovering suffixes: A Case Study for MarathiLanguage" (IJCSE) International Journal on Computer Science and Engineering Vol. No. 02, Issue No. 08, pp 2716-2720.

[13] Haidar Harmani, Walid Keirouz, & Saeed Raheel (2006) "A rule base extensible stemmer for Information retrieval with application to Arabic" The international Arab journal of information technology, Vol No.3, Issue No.3, pp 265-272.

[14] Abduelbaset M. GOoweder, Husien A. Alhammi , Tarik Rashed,and Abdulsalam Musrati " A Hybrid Method for Stemming Arabic Text " Journal of computer Science,URL:http://eref.uqu.edu.sa/files/eref2/folder6/f181.pdf.

[15] Georgios Ntais (2006)"Development of a Stemmer for the Greek Language"Department of Computer and Systems SciencesMaster Thesis at Stockholm University / Royal Institute of Technology, pp1-40.

[16] www.wikipedia.com

[17] M.F Porter (1980)" An algorithm for suffix stripping" Published in Program, Vol No.14, Issue No.3, pp 130- 137,URL:http://www.cs.odu.edu/~jbollen/IR04/readings/readings5.pdf.

[18] Jörg Caumanns (1998) “A Fast and Simple Stemming Algorithm for German Words1”Algorithm is Publish in Department of computer science at the free university of Berlin, pp 1-10,

[19] Bal Krishna Bal, Prajol Shrestha (2004)”A Morphological Analyzer and a Stemmer for Nepali” PAN Localization, Working Papers 2004-2007, pp 324-31.

[20] Md. Zahurul Islam, Md. Nizam Uddin and Mumit Khan (2004)” A Light Weight Stemmer for Bengali and Its Use in Spelling Checker” Proceedings of 1st International Conference on Digital Communications and Computer Applications (DCCA2007), Irbid, Jordan, pp 87-93.

[21] Dinesh Kumar and Prince Rana (2010) “Design and Development of Stemmer for Punjabi”, International Journal of Computer Applications (0975 – 8887)