AN UNSUPERVISED APPROACH TO DEVELOP STEMMER

Mohd. Shahid Husain

Department of Information Technology, Integral University, Lucknow
siddiquisahil@gmail.com

ABSTRACT

This paper presents an unsupervised approach for the development of a stemmer (For the case of Urdu & Marathi language). Especially, during last few years, a wide range of information in Indian regional languages has been made available on web in the form of e-data. But the access to these data repositories is very low because the efficient search engines/retrieval systems supporting these languages are very limited. Hence automatic information processing and retrieval is become an urgent requirement. To train the system training dataset, taken from CRULP [22] and Marathi corpus [23] are used. For generating suffix rules two different approaches, namely, frequency based stripping and length based stripping have been proposed. The evaluation has been made on 1200 words extracted from the Emille corpus. The experiment results shows that in the case of Urdu language the frequency based suffix generation approach gives the maximum accuracy of 85.36% whereas Length based suffix stripping algorithm gives maximum accuracy of 79.76%. In the case of Marathi language the systems gives 63.5% accuracy in the case of frequency based stripping and achieves maximum accuracy of 82.5% in the case of length based suffix stripping algorithm.

KEYWORDS

Stemming, Morphology, Urdu stemmer, Marathi stemmer, Information retrieval

1. INTRODUCTION

More and more information is now being made available online. English and European Languages basically dominated the web since its inception. However, now the web is getting multi-lingual. Especially, in the past few years, there has been a significant increase in amount of information in Indian and other Asian languages. Document in a large number of Indian languages like Hindi, Urdu, Bengali, Oriya, Tamil, Telugu and Marathi are now available in the electronic form. Information retrieval (IR) systems play a vital role in providing access to this information. A number of information retrieval systems are available in English and other European languages.

Work involving development of IR systems for Indian languages is only of recent interests. Development of such systems is constraint by the lack of the availability of linguistic resources and tools in these languages. Stemmer is one such tool, which almost all of the IR systems use to reduce morphological variants of a word to its root or stem. The stem is not necessarily the linguistic root of the word.

DOI : 10.5121/ijnlc.2012.1202

For example, words like भारताची, भारतासाठा, भारतामध्य, भारतानी after stemming are mapped to common word form as भारत Similarly the root word of آخری is آخر (without the ی).

Stemming offers two important benefits to an IR system. First, it increases the recall of the system since the query words are matched with their morphological variants in the documents and second, it reduces the index size thereby leading to significant benefits in speed and memory requirements. Standard stemmers are available for English. However, no such stemmers are available for most of the Indian languages, including Marathi and Urdu. Two most widely used stemmers for English, namely Porter's [9] and Lovin's [4] stemmer are rule-based. Rule-based stemmers make use of hand-crafted linguistic rules. Obtaining such rules is difficult and time consuming besides being language specific.

In this paper, we present an unsupervised approach for the development of a stemmer for Urdu and Marathi language, languages widely spoken inIndia. Unsupervised approach learns suffixes automatically from a large vocabulary of words extracted from raw text. Two different stripping methods have been investigated. To the best of our knowledge no published literature reports on unsupervised stemming in the case of Urdu and Marathi language. Marathi is morphologically very rich. Marathi words may have many morphological variants mostly formed by adding suffixes like ◌ ◌◌ा, ◌◌ी, ◌◌ामुळे, ◌◌ामधील and vice versa to the same stem. The words like मुलाला, मुलानी, मुलांमुळे and मुलांमधील are the morphological variants of common word मुल Similarly Urdu is also a morphological rich language. For example the stem for the words آخری and آخرکار is آخر.

The main problem in IR field for Indian languages is that India is a multilingual country having more than 20 regional languages. So for Indian context, the IR approach used should be capable of handling multilingual documents. To access information available in English or some other European languages there are number of efficient IR systems. Work involving development of IR systems for Asian languages is only of recent interests. Development of such systems is constraint by the lack of the availability of linguistic resources and tools in these languages. The same problem is with the Indian languages. Stemmer is one such tool, which almost all of the IR systems use.

Stemming is the backbone process of any IR system. Stemmers are used for getting base or root form (i.e. stems) from inflected (or sometimes derived) words. Unlike morphological analyzer, where the root words have some lexical meaning, it's not necessary with the case of a stemmer. Stemming is used to reduce the overhead of indexing and to improve the performance of an IR system. More specifically, stemmer increases recall of the search engine, whereas Precision decreases. However sometimes precision may increases depending upon the information need of the users. Stemming is the basic process of any query system, because a user who needs some information on آخری may also be interested in documents that contain the word آخر (without the ی).

The approaches used for developing a stemmer can be broadly classified as Rule-based (knowledge-based) and machine learning (supervised and unsupervised) approaches. A rule based stemmer makes use of linguistic knowledge to develop rules for stemming. Besides being language specific it is very difficult and time consuming to obtain such rules. Specifically, for languages like Urdu, which is a very highly inflectional language, the task becomes quite cumbersome.

Supervised learning is an alternative approach to frame stemming rules. In order to learn suffixes this approach uses set of inflection-root pair of words which are manually segmented. But this algorithm is also not produce very effective results for Urdu language as it is highly inflectional language and this becomes a complex task. Manually segmenting the Urdu words is a very time-consuming task and is not feasible because in Urdu for a root word there are many inflections. It also requires a very good linguistic knowledge to segment words and get the root and the inflections.

For designing stemmer we have used unsupervised stemming approach. This approach does not require any specific knowledge of the language in case. It uses a set of words (training dataset) to learn suffixes. As the approach used in this work is language independent, it can be easily used for the development of the stemmers of other languages as well. For suffix rule generation two different approaches have been discussed. First is the Length based approach which is very simple suffix stripping approach. The second is Frequency based approach. The experiment results shows that the second approach used, gives the more accurate results. The rest of the paper is organized as follows:

Section 2 reviews the earlier work done in morphological analysis and stemming for Indian languages. Section 3 gives a brief idea about the proposed approach. Section 4 presents the detail of experimental setup. Section 5 discusses the important results and observations and finally conclusion have been made in section 6.

2. LITERATURE SURVEY

The most basic component of any Information Retrieval system is Stemmers. Among all the morphological systems, stemmers are the simplest system. Since the very beginning of the Information Retrieval era, the focus of the IR community is on developing efficient stemming algorithms to support various languages. Earlier most of the work done for stemmers was mainly rule based. The rule Formulation based on the linguistic inputs is a very complex and tough job. Moreover it requires very good linguistic knowledge to design such stemmers. Also Rule based stemmers are language specific and thus it constraint its use. Hence we have to develop universal stemmers, which can be used for any language and have no linguistic constraint.

Earlier stemmers were designed on rule-based approach. Julie Lovins published the first paper on rule-based stemming in the year 1968. In this about 260 rules were listed for stemming the English language. The approach used by Lovins was Iterative Longest Match heuristic. The most noteworthy work in the field of rule based stemmer was presented by Martin Porter in 1980 [9]. He simplified the rules of Lovin to about 60 rules. The algorithm developed by him is called Porter Stemmer. This is very simple and effective algorithm and is widely used to develop search engines. To access information available in English or some other European languages there are number of efficient IR systems. Work involving development of IR systems for Asian languages is only of recent interests. Development of such systems is constraint by the lack of the availability of linguistic resources and tools in these languages. The same problem is with the Indian languages.

Until recently, For Indian regional languages the work done by IR community involves languages like Hindi, Bengali, Tamil and Oriya. But there is no reported work done for Urdu and Marathi language. Although, as per our knowledge there is no reported works done by the IR community to efficiently retrieve the information available on net in Urdu language, however, a lot of research has been done towards computational morphological analysis and stemming of Urdu. Computational analysis of different parts of speech in Urdu is described by Rizvi [1] and Butt [2].

To stem French words in a corpus a dictionary-based approach is used [3]. Various researches have been done on Arabic and Farsi stemmers, most of them uses statistical and heuristics based approaches [4, 5]. A supervised approach was proposed by R. Wicentowski [6] for learning the set of suffixes for the purpose of stripping the word and get root word. This approach was based on the Word Frame model. In this approach a set of inflection-root pairs was used to train the stemmer. A comprehensive computational analysis of Urdu morphology is given by Hussain [7].

Stemmers may be developed by using either rule based or statistical approaches. Rule based stemmers require prior morphological knowledge of the language, while statistical stemmers use corpus to calculate the occurrences of stems and affixes. Both rule-based and statistical stemmers have been developed for a variety of languages. A rule-based stemmer is developed for English by Krovetz, using machine-readable dictionaries. Along with a dictionary, rules for inflectional and derivational morphology are defined. Due to high dependency on dictionary the systems lacks consistency [8]. To perform stemming of Arabic an approach using stop word list is proposed by Thabet. This algorithm gives accuracy of 99.6% for prefix stemming and 97% for postfix stemming [10]. Paik and Parui [11] have proposed an interesting stemming approach based on the general analysis of Indian languages. For Persian language a rule based algorithm was proposed by Sharifloo and Shamsfard for stemming this algorithm uses bottom up approach. The accuracy of this algorithm is 90.1 % [12]. Besides rule-based stemmers there are a number of statistical stemmers for different languages. These stemmers use some statistical analysis of the training data and then rules are derived from these analyses for stripping the inflected words to get the root word. Croft and Xu provide two methods for stemming i.e. Corpus-Specific Stemming and Query-Specific Stemming [13]. Kumar and Siddiqui propose an algorithm for Hindi stemmer which achieves 89.9% accuracy [14]. For stemming Telugu language, Murthy et.al. Present's three statistical techniques. This approach increases accuracy to 74.5% [15]. Instead of the conventional lexical lookup employed for developing stemmers in other languages an Urdu stemmer called Assas-Band, has been developed by Qurat-ul-Ain Akram, Asma Naseer, Sarmad Hussain (Assas-Band, an Affix-Exception-List Based Urdu Stemmer) using more precise affix based exception lists, which increases accuracy up to 91.2% [16].

Earlier work in Indian Morphology includes [12, 13, 14, and 15]. Larkey et al. [12] used a light weight stemmer that uses a manually formulated list of 27 most common suffixes for stemming. A similar approach was proposed by Ramanathan and Rao [13]. They used a manually extractied list of 65 most inflectional suffixes. A statistical Hindi stemmer was developed by Chen and Grey [14] for evaluating the performance of the Hindi information retrieval System. Similar work has been done by Dasgupta and Ng [15] for Bengali Morphological Analyzer. An approach based on "Observable Paradigms" for Hindi Morphological Analyzer is proposed in [16]. A rule based approach TelMore was proposed by [17] for telugu language.

3. APPROACH USED

Our proposed approach is based on n-gram splitting model. For learning purpose of the stemmer, documents from the Urdu Corpus available at CRULP are used. The words taken from these documents are split to get n split suffixes, using n gram model. Where n=1, 2, 3…l, for word length l.

Then the frequency count of the split words is calculated to get the probability of the stem - suffixes pair extracted from the n-gram splitting.

Then we have calculated the optimal split probability, which is the multiplication of the stem probability and suffix probability. By observing the results, a particular frequency threshold was taken. The splits whose frequency count lies above this threshold value were considered as valid candidates and were used for suffix generation rules. Also the maximum split probability corresponds to the optimal split segments which are considered to be the valid candidate for framing suffix generation rule.

Table 1: Algorithmic steps

- Split words into n gram
- Generate stem and suffix list
- Sort suffixes on decreasing order of their frequency
- Generate suffix stripping rules
  - i. using Frequency based stripping
  - ii. using length based stripping

3.1 Word splitting and Stem Classes generation

In this step n-gram model is used to obtain corresponding stems and suffixes of a word Wy by splitting it into n-grams as given below
Wy: = {(stem$_{1y}$|suffix$_{1y}$); (stem$_{2y}$|suffix$_{2y}$); …... (stem$_{xy}$|suffix$_{xy}$)}
Where x, y=1, 2, 3… l (where l denotes the length of the word) and stem$_{xy}$ is the xth stem of yth word and suffix$_{xy}$ is the xth suffix of yth word.
For example, the word آئزلینڈ gives the following stem-suffix pairs after n-gram splitting:
آئزلینڈ := { (NULL -- آئزلینڈ); (آ ئزلینڈ); (آئ زلینڈ); (آئز لینڈ); (آئزل ینڈ); (آئزلی نڈ); (آئزلین ڈ); (آئزلینڈ --NULL) }

For example, the word अंकात can be split up into following stems and suffixes:

अंकात := { (अंकात | NULL); (अंका | त); (अंक | ात); (अं | कात); (अ | ंकात);

(NULL| अंकात) }

Next a common stem class is used to group the words having common stems. To find common stems, maximum common prefix method is used.
For example the stem equivalence class for the words آخری and آخرکار
Can be given as: آخری = :{آخر, آخرکار}

For example the words िचऽपटांतील, िचऽपटाचा, िचऽपटाचे, िचऽपटात, िचऽपटाने can be grouped under the stem class िचऽपट as:

िचऽपट := [िचऽपटांतील, िचऽपटाचा, िचऽपटाचे, िचऽपटात, िचऽपटाने]

3.2 Generation of Stem and Suffixes

The longest common prefix method is used to obtain the correct stems and suffixes from the inflected words. We have used the stem equivalence class, generated in the first phase of the algorithm to find out the longest common prefixes. These prefixes are then stored as the stems and the remaining part of the word as the valid suffix along with its corresponding frequency count. This information is then used to frame rules for suffix stripping. The suffixes in the generated list having higher frequency are considered as valid suffixes for generating suffix stripping rule. For example the common root word of different inflected words with their suffixes is stored as; {کار ,ی} = :آخر

3.3 Frequency Counting

In this step the frequency count of the suffixes generated in step 2 is calculated. This list of suffixes is then arranged in order of their count. By manual analyses of the system a frequency count is taken as the threshold. The suffixes having there frequency count below this threshold value are discarded and not considered for suffix rule generation while those lying above the preset threshold value are considered as the valid candidates for framing the suffix stripping rules.

3.4 Generation of Suffix Rules

In this step, two different approaches are used for the purpose of suffix stripping rule generation.

3.4.1 Length Based Suffix Stripping

This is the crudest method for suffix rule generation. In this approach, the suffix list obtained from step 2 is sorted according to their lengths in decreasing order. This approach is quite valid as it removes the suffix in a word which is of max length. The drawback of this approach is that in many cases over-stemming occurs.

3.4.2 Frequency Based Suffix Stripping

This is the simplest method for generating suffix stripping rule. The suffixes obtained in the second step, are sorted in descending order of their corresponding frequency counts. By manual observation a threshold value is being set. The suffixes having there frequency count below this threshold value are discarded and not considered for suffix rule generation while those lying above the preset threshold value are considered as the valid candidates for framing the suffix stripping rules. This method is quite effective for Urdu and other very highly inflectional languages because as they have very large number of suffixes.

4. EXPERIMENT

For the evaluation purpose of the proposed stemmer, following experiment was conducted. The parameter used to measure the performance of the stemmer is accuracy. The accuracy can be defined as the fraction of words stemmed correctly. Mathematically it can be stated as:

$$\text{Accuracy} = \frac{\text{Number of Correctly stemmed Words}}{\text{Total Number of Words}} \text{ x } 100$$

For testing of the stemmer a list of 1200 words, taken from Emille corpus, with their suffixes and stems is created manually. Then the developed system is used to get the stem of these words and cross checked with the list of manually stemmed words. The following table gives a summary about the statistics used for the evaluation of the stemmer.

4.1 FOR URDU LANGUAGE

Table 2 Data Set Specification

Training Dataset	D1	D2	D3
Source	EMILLE corpus	EMILLE corpus	EMILLE corpus

Count of words	50495	50836	10559
Count of Unique words	6428	6178	2492
Test Data set	1200	1200	1200

To perform the evaluation of the proposed stemmer, the experiment is conducted in three runs. In Run1 Dataset D1 have been used for training, in Run2 Dataset D2 have been used for training and in Run3 Dataset D3 have been used for training. The statistics used for evaluation are shown in the following table.

Table 3 Experiment Specification

Run	R1	R2	R3
Training Dataset	D1	D2	D3
Testing Dataset	Test Dataset	Test Dataset	Test Dataset

Table 4 Results of the Experiments

Run	Accuracy of Implemented approach	
	Frequency based	Length based
R1	82.78	81.28
R2	85.36	77.85
R3	84.67	79.76

Table 4 shows the comparison between results obtained by using different methods for Urdu Language.

4.2 FOR MARATHI LANGUAGE

Table 5 Data Set Specification

Training Dataset	D1	D2	D3
Source	Marathi corpus	Marathi corpus	Marathi corpus
Count of words	132895	100836	106559
Count of Unique words	27613	16178	12492
Testing words	1200	1200	1200

To perform the evaluation of the proposed stemmer, the experiment is conducted in three runs. In Run1 Dataset D1 have been used for training, in Run2 Dataset D2 have been used for training and in Run3 Dataset D3 have been used for training. The statistics used for evaluation are shown in the following table.

Table 6 Experiment Specification

Run	R1	R2	R3
Training Dataset	D1	D2	D3
Testing Dataset	Test Dataset	Test Dataset	Test Dataset

Table 7 Results of the Experiments

Run	Accuracy of Implemented approach	
	Frequency based	Length based
R1	63.52	71.37
R2	57.29	82.68
R3	61.82	79.76

Table 7 shows the comparison between results obtained by using different methods for Marathi Language.

5. RESULTS AND DISCUSSION

It is clear from table 4, that the frequency based suffix generation approach gives the maximum accuracy of 84.27% whereas Length based suffix stripping algorithm gives maximum accuracy of 79.63% in case of Urdu language. For Marathi language the system gives 63.5% accuracy in frequency based approach and achieves maximum accuracy of 82.68% in length based suffix stripping approach.

Effect of stop words on stemming: when we have removed the stop words from the training dataset then there is some effect on the suffix list generated (the number of suffixes decreases by 2%), but there is no effect on stemming i.e. the result of stemmer is same after the stop word removal as it was before the stop word removal.

6. CONCLUSION AND FUTURE WORK

The approach used in this work gives promising results for Urdu as well as Marathi language. As the approach used is language independent it can be tested and implemented for other languages in near future.

As there is some problem of under stemming and over stemming in the used approaches. In future one can attempt to reduce these effects to improve the efficiency of the system.

As we know that stemmers have tremendous use in the Information Retrieval. We plan to make use of the designed stemmer for other related work of Information retrieval.

REFERENCES

[1] Rizvi, J et. al. “Modeling case marking system of Urdu-Hindi languages by using semantic information”. Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE '05). 2005.

[2] Butt, M. King, T. “Non-Nominative Subjects in Urdu: A Computational Analysis”. Proceedings of the International Symposium on Non-nominative Subjects, Tokyo, December, pp. 525-548, 2001.

[3] Savoy, J. “Stemming of French words based on grammatical categories”. Journal of the American Society for Information Science, 44(1), 1-9, 1993.

[4] Lovins Julie Beth: Development of a stemming algorithm. Mechanical
Translation and Computational Linguistics 11:22–31. (1968)

[5] Mokhtaripour, A., Jahanpour, S. “Introduction to a New Farsi Stemmer”. Proceedings of CIKM Arlington VA, USA, 826-827, 2006.

[6] R. Wicentowski. "Multilingual Noise-Robust Supervised Morphological Analysis using the Word Frame Model." In Proceedings of Seventh Meeting of the ACL Special Interest Group on Computational Phonology (SIGPHON), pp. 70-77, 2004.
[7] Rizvi, Hussain M. “Analysis, Design and Implementation of Urdu Morphological Analyzer”. SCONEST, 1-7, 2005.
[8] Krovetz, R. “View Morphology as an Inference Process”. In the Proceedings of 5th International Conference on Research and Development in Information Retrieval, 1993.
[9] Porter, M. “An Algorithm for Suffix Stripping”. Program, 14(3): 130-137, 1980.
[10] Thabet, N. “Stemming the Qur’an”. In the Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, 2004.
[11] Paik, Pauri. “A Simple Stemmer for Inflectional Languages”. FIRE 2008.
[12] Sharifloo, A.A., Shamsfard M. “A Bottom up Approach to Persian Stemming”. IJCNLP, 2008
[13] Croft and Xu. “Corpus-Based Stemming Using Co occurrence of Word Variants”. ACM Transactions on Information Systems (61-81), 1998.
[14] Kumar, A. and Siddiqui, T. “An Unsupervised Hindi Stemmer with Heuristics Improvements”. In Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data, 2008.
[15] Kumar, M. S. and Murthy, K. N. “Corpus Based Statistical Approach for Stemming Telugu”. Creation of Lexical Resources for Indian Language Computing and Processing (LRIL), C-DAC, Mumbai, India, 2007.
[16] Qurat-ul-Ain Akram, Asma Naseer, Sarmad Hussain. “Assas-Band, an Affix-Exception-List Based Urdu Stemmer”. Proceedings of ACL-IJCNLP 2009.
[17] http://en.wikipedia.org/wiki/Urdu
[18] .http://www.bbc.co.uk/languages/other/guide/urdu/steps.shtml
[19] http://www.andaman.org/BOOK/reprints/weber/rep-weber.htm
[20] Natural Language processing and Information Retrieval by Tanveer Siddiqui, U S Tiwary.
[21] Information retrieval: data structure and algorithms by William B. Frakes, Ricardo Baeza-Yates.
[22] http://www.crulp.org/software/ling_resources.htm
[23] Marathi Corpus, http://www.cfilt.iitb.ac.in/marathi_Corpus/ , IIT Powai, Mumbai

Authors

Mohd. Shahid Husain

M.Tech. from Indian Institute of Information Technolo gy (IIIT-A), Allahabad with Intelligent System as specialization. Currently pursuing Ph.D. and working as assistant professor in the department of Information Technology, Integral University, Lucknow.