RESEARCH ARTICLE OPEN ACCESS

Unsupervised Kannada Stemmer using Partial Lemmatizer and Indo – WordNet

B H Manjunatha Kumar*, Dr.M.Siddappa**, Dr.J.Prakash***

**(Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Tumkur, India)*
*** (Professor and Head, Department of Computer Science and Engineering, Siddhartha Institute of Technology, Tumkur, India)*
**** (Professor and Head, Department of Information Science and Engineering, BIT, Bengaluru, India)*

ABSTRACT
Stemming is basically an operation that converts morphologically related words to a common stem or root word by removing their suffixes or prefixes, but it is not necessary that root or stem is a proper dictionary word. The algorithm proposed in this paper overcome this problem by embedding partial lemmatizer with stemmer. It also makes use of Indo WordNet to determine the proper root word. An unsupervised approach is used in this algorithm and it is implemented for the Kannada language. Since it is an unsupervised method, so this idea can be used to design stemmer for other than the Kannada language also.

Keywords - Lemmatizer, Stemming, Unsupervised stemmer, WordNet

Date of Submission: 11-12-2017 Date of acceptance: 22-12-2017

I. INTRODUCTION

In the past few years much and more information is being made available online in Indian and other Asian languages. A large number of documents in Indian languages is now available in electronic form. Information Retrieval (IR) system plays an important role in accessing this information. The development of IR systems for Indian languages is constraint by the lack of availability of linguistic resources and tools in these languages. Almost all of the IR systems use the tool stemmer for reducing morphological variants of an inflected word to its stem or root word.

Stemming and Lemmatization are two important basic Natural Language Processing (NLP) techniques used in many NLP applications. Language has its own grammar and it is so vast, so it is absolutely needed to study the structure of the word. The stemmer objective is to converting morphologically equal words into root word without doing the morphological analysis of that word. To understand the internal structure of the derived words of the language morphological analyzer is very much required. To design a morphological analyzer and generator for any language the basic information required is the stems and suffixes. For example, an ideal stemmer should normalize the words *produce, producing, produced, production* to the stem *produc*. Here the word produc is not a proper root word. Lemmatizer will give a solution to this problem by detaching the inflectional ending and generates the base or dictionary form of the word. The root word is searched in the Indo WordNet if it present in the Indo WordNet then that root word is considered as the genuine root word of the derived terms.

There are mainly two approaches to develop the stemmer, they are named as Rule-based (Knowledge-based) and Machine learning (Supervised or Unsupervised) approaches.

In rule-based stemmer, linguistic knowledge is used to develop the rules for stemming. The development of such rules is very difficult and time-consuming. The language like Kannada, which is very highly inflectional language, the task is quite cumbersome.

In Supervised learning approach set of manually segmented inflectional – root pair of words are used to learn suffixes. But this approach is also a complex task and time consuming for highly inflectional language like Kannada. This approach also requires a very strong linguistic knowledge to segment words to get the root and the inflections.

In this approach, we have used unsupervised stemming approach. It doesn’t require any specific knowledge of the language. Suffix stripping approach is used for suffix rule generation.

The paper is organized as follows: Section 2 reviews the earlier work done in stemming for Indian languages. The proposed algorithm with sample examples is dealt in section 3. Section 4 is dedicated to explaining the results and analysis. Conclusion and directions for future work are posted in section 5.

II. RELATED WORK

For English and other European languages, many stemming and lemmatization algorithms have already been developed. Amaresh Kumar Pandey et al. [1] proposed an unsupervised stemmer for the Hindi language with heuristic improvements. It uses split all method to form the stem. Ramanathan and Rao developed a lightweight stemming algorithm for Hindi [2]. It uses a handcrafted rule-based approach for stemming. This paper introduces a suffix stripping approach for a noun, adjective and the verb inflections occurring in the Hindi language.

Dinesh Kumar and Prince Rana proposed brute force technique for stemming Punjabi words [3]. It uses a lookup table which contains the relationship between the root forms and inflected forms. In this paper, the table is used to derive the matching inflection to stem a word. If a matching inflection is found then the root form is returned.

Prasenjit Majumder et al. proposed Yet Another Suffix Stripper (YASS) [4] and tested for English, Bengali and French. It uses unsupervised classification of data into various clusters. In this paper, four distance measures for clustering the words are proposed. It uses complete linkage clustering algorithm belongs to the class of hierarchical clustering for clustering the words. The aim of this work was to improve the accuracy of IR.

KVN Sunitha et al. (2009) proposed an approach to improving word coverage using unsupervised morphological analyzer [5]. This approach combines statistical approach and clustering to improve stemming. This algorithm breaks a word in all possible locations based on maximum suffix length and minimum stem length and filters out the important decompositions on the basis of the frequency of occurrence. K – mean algorithm is used for clustering.

Deepa Gupta et al. proposed an approach for improving unsupervised stemming by using partial lemmatization for the Hindi language [6].

Rajeev Puri et al. proposed Punjabi stemmer using Punjabi WordNet database [7]. In this paper, a revised suffix removal approach with an extended set of stripping rules has been discussed for creating a Punjabi language Stemming tool. The stemming algorithm discussed in this paper uses regular expressions for finding suffix matches. The WordNet database is used here for improving the stemming results.

Rohit Kansal et al. proposed Rule Based Urdu Stemmer [8]. In this paper rules are applied to remove suffix and prefix from the inflected words.

Snigdha Paul et al. proposed a rule based hindi lemmatizer [9]. In this paper an inflectional lemmatizer is designed, which generates the rules for extracting the suffixes and also added rules for generating a proper meaningful root word.

III. PROPOSED ALGORITHM AND ILLUSTRATION

As stated earlier, in this paper we attempt to improve the quality of stemming using lemmatization and Indo-WordNet.

To get a better understanding of the algorithm, the steps are illustrated with some of the sample outputs.

Step 1: Pre-processing

This step involves two operations: first one is Normalise the input text by removing the special characters like ! , ; : ( ) { } [ ] \ / + - % ? and the second one is convert the different encoded input text to a common encoding scheme like UTF8.

Step 2: Create a word list by tokenizing the input text

Tokenize the input text to create a word list. This step essentially requires feeding a raw text to generate a word list.

Step 3: Generate all possible stems and suffixes

Each word of the word list is decomposed into all possible suffixes and stems. The maximum length of the suffix and minimum length of the stem is assumed as 8 and 2 respectively.

For the word ಪ್ರಯೋಜನಕಾರಿ, the stems and suffixes are shown in below Table 1:

Table 1 Output of step 3

Word	Stem	Suffix
ಪ್ರಯೋಜನಕಾರಿ	ಪ್ರಯ	◌ೋಜನಕಾರಿ
	ಪ್ರಯೋ	ಜನಕಾರಿ
	ಪ್ರಯೋಜ	ನಕಾರಿ
	ಪ್ರಯೋಜನ	ಕಾರಿ
	ಪ್ರಯೋಜನಕ	◌ಾರಿ
	ಪ್ರಯೋಜನಕಾ	ರಿ
	ಪ್ರಯೋಜನಕಾರ	◌ಿ

Step 4: Collect all the suffixes that occur with the similar stem

For the word set { ಪ್ರಯೋಜನೆ, ಪ್ರಯೋಜನಗಳು, ಪ್ರಯೋಜನವಿಲ್ಲ, ಪ್ರಯೋಜನವಾಗಲಿಲ್ಲ} output is shown in below Table 2:

Table 2 Output of step 4

Stem bag	Suffix bag
ಪ್ರಯೋಜನೆ	@
ಪ್ರಯೋಜನಗಳು	@
ಪ್ರಯೋಜನವಿಲ್ಲ	@
ಪ್ರಯೋಜನವಾಗಲಿಲ್ಲ	@

ಪ್ರಯೋಜನಗಳ	◌ು
ಪ್ರಯೋಜನವಿ	ಲ್ಲ
ಪ್ರಯೋಜನವಾ	ಗಲಿಲ್ಲ
ಪ್ರಯೋಜನ	ಗಳು, ವಿಲ್ಲ, ವಾಗಲಿಲ್ಲ
ಪ್ರಯೋಜನವಾಗಲಿ	ಲ್ಲ
ಪ್ರ	ಯೋಜನೆ, ಯೋಜನಗಳು, ಯೋಜನವಿಲ್ಲ, ಯೋಜನವಾಗಲಿಲ್ಲ
ಪ	◌್ರಯೋಜನೆ, ◌್ರಯೋಜನಗಳು, ◌್ರಯೋಜನವಿಲ್ಲ, ◌್ರಯೋಜನವಾಗಲಿಲ್ಲ
ಪ್ರಯೋಜನವಾಗಲ	◌ಿಲ್ಲ
ಪ್ರಯ	◌ೋಜನೆ, ◌ೋಜನಗಳು, ◌ೋಜನವಿಲ್ಲ, ◌ೋಜನವಾಗಲಿಲ್ಲ
ಪ್	ರಯೋಜನೆ, ರಯೋಜನಗಳು, ರಯೋಜನವಿಲ್ಲ, ರಯೋಜನವಾಗಲಿಲ್ಲ
ಪ್ರಯೋ	ಜನೆ, ಜನಗಳು, ಜನವಿಲ್ಲ, ಜನವಾಗಲಿಲ್ಲ
ಪ್ರಯೋಜನವಾಗ	ಲಿಲ್ಲ
ಪ್ರಯೋಜನವ	◌ಿಲ್ಲ, ◌ಾಗಲಿಲ್ಲ
ಪ್ರಯೋಜನಗ	ಳು
ಪ್ರಯೋಜನವಾಗಲಿಲ್	ಲ
ಪ್ರಯೋಜನವಿಲ	◌್ಲ
ಪ್ರಯೋಜ	ನೆ, ನಗಳು, ನವಿಲ್ಲ, ನವಾಗಲಿಲ್ಲ
ಪ್:	ರಯೋಜನೆ, ರಯೋಜನಗಳು, ರಯೋಜನವಿಲ್ಲ, ರಯೋಜನವಾಗಲಿಲ್ಲ

Step 5: Eliminate all stems whose size is less than two.

Stems of size less than two have eliminated as shown in below Table 3.

Table 3 Output of step 5

Stem Bag	Suffix Bag
ಪ	◌್ರಯೋಜನೆ, ◌್ರಯೋಜನಗಳು, ◌್ರಯೋಜನವಿಲ್ಲ, ◌್ರಯೋಜನವಾಗಲಿಲ್ಲ

Step 6: Cluster the Stems having the similar suffix set. The output is as shown in below Table 4.

Table 4 Output of step 6

Stem bag	Suffix bag
ಅಣ್ಣ , ತಮ್ಮ	◌ಂದಿರು, ◌ಂದಿರನ್ನು, ◌ಂದಿರಿಗೆ

Step 7: Lemmatization. Output of lemmatization process is as shown in below Table 5

Table 5 Output of step 7

Stem bag	Suffix bag
ಪ್ರಯೋಜನ	◌ೆ, ಗಳು, ವಿಲ್ಲ, ವಾಗಲಿಲ್ಲ
ಅಣ್ಣ , ತಮ್ಮ	◌ಂದಿರು, ◌ಂದಿರನ್ನು, ◌ಂದಿರಿಗೆ

IV. Results and Analysis

The proposed algorithm is implemented using Python programming language and run over the randomly chosen dataset of 50 Kannada words. We manually verified the output results using the effectiveness evaluation measures viz. Recall, Precision and F-score. These measures are computed as:

%Recall (R) =

$$\frac{(total\ number\ of\ correct\ decomposition\ given\ by\ system * 100)}{(Total\ number\ of\ correct\ decomposition)}$$

%Precision (P) =

$$\frac{(Total\ no\ correct\ decomposition\ given\ by\ system * 100)}{(Total\ number\ of\ decompostiom\ give\ by\ the\ system)}$$

F – Score $= \frac{2 * R * P}{(R + P)}$

The values of the above three metrics achieved when the test data set was run through the proposed system are Recall(R) = 88%, Precision(P) = 77.19% and F – Score = 82.24%.

V. Conclusion

In this paper, we have discussed the development of Kannada stemmer. The work focuses on unsupervised approach. The main aim is to improve the stemming by using partial lemmatization and Indo – WordNet. The concept of making use of partial lemmatization and Indo - WordNet in unsupervised stemming which has never been explored for the Kannada language up to now.

.

REFERENCES

[1] Amaresh Kumar Pandey, Tanveer J Siddiqui, 2008, An unsupervised Hindi Stemmer with heuristic improvements, In Proceedings of the second workshop on Analytics for noisy unstructured text data, 2008, pp 99-105, Singapore.

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

[3] Dinesh Kumar and Prince Rana, 2001, Stemming of Punjabi words by using brute force technique, International Journal of Engineering Science and Technology (IJEST).

[4] Prasenjit, Majumder., et al., 2007, YASS: Yet another suffix stripper. ACM Transactions on Information Systems. 25 (4) Article No. 18

[5] KVN Sunitha and N Kalyani, Sadhana, 2009 Improving Word Coverage using unsupervised morphological analyzer, Indian Academy of science

[6] Gupta, Deepa Yadav, Rahul Kumar Sajan, Nidhi, 2012, Improving Unsupervised Stemming by using Partial Lemmatization Coupled with Data-based Heuristics for Hindi, International Journal of Computer Applications (0975 – 8887) Volume 38– No.8, January 2012, Pages 1 – 8.

[7] Rajeev Puri et al., 2015, Punjabi stemmer using Punjabi WordNet database, Indian Journal of Science and Technology, Vol 8(27), October 2015.

[8] Rohit Kansal, Vishal Goyal and G. S. Lehal, 2012, Rule Based Urdu Stemmer, Proceedings of COLING 2012: pages 267–276, Mumbai.

[9] Snigdha Paul, Mini Tandon, Nisheeth Joshi and Iti Mathur, Design Of A Rule Based Hindi Lemmatizer, ACITY, AIAA, CNSA, DPPR, NeCoM, WeST, DMS, P2PTM, VLSI - 2013 pp. 67–74, 2013. © CS & IT-CSCP 2013.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

B H Manjunatha Kumar "Unsupervised Kannada Stemmer using Partial Lemmatizer and Indo – Word Net." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 12, 2017, pp. 39-42.