Stemming Techniques and Naïve Approach for Gujarati Stemmer

Jikitsha R. Sheth
Assistant Professor
SRIMCA, Uka Tarsadia University
Dist. Surat, Gujarat, India

Bankim C. Patel, Ph.D.
Director
SRIMCA, Uka Tarsadia University
Dist. Surat, Gujarat, India

ABSTRACT
This paper focuses on stemming techniques used by various researchers for English language. Further the review of different approaches used for Hindi and Bengali languages are discussed. The authors have proposed a model for Gujarati stemmer by incorporating traditional as well as naïve approaches.

General Terms
Morphemes, Stemmer, Indian languages, Gujarati

1. INTRODUCTION
Morphemes are the smallest meaningful units that make up words. The stems bear the core meaning while the affixes are the units that join with the stem most of the time as a grammatical function. The process of stemming implies the reduction of a term to its stem. Generally it is done by stripping off the affixes i.e. suffixes and prefixes. This process is language dependent. For example, in English language the words compensate or compensation will get reduced to compen. In Gujarati language the word કરવું(to do) and કરાવવું(to make someone do) will get reduced to કર(do).

Information retrieval deals with the organization, storage, retrieval, and evaluation of information relevant to a user's query. The retrieval system returns those documents which match the query of user. Most of the time the query of user is in natural language and the relevant documents may have conflation or morphological variants of the searched terms. For this stemming can help by reducing the terms of user query to their respective stem and match that stem in the documents to find the relevancy.

2. STEMMING TECHNIQUES
Lovins[22] suggested rule based stemming approach by providing 260 suffixes list. Affixes removal stemming techniques were given by Porter [16] and Paice[17] for English language. Such stemming techniques are also known iterative and/or longest match stemmer that are based on rules like replace the occurrence of suffix 'sses' with 'ss', 'ies' with i, 's' with null string, 'ing' with null string and so on. In nineties, stem tables concept was proposed by Frakes and Yates[14] to create tables that store all the possible index terms and their corresponding stems as shown in table 1.

Table 1: Stem tables

Terms	Stem
Singing	Sing
Sang	Sing
Sung	Sing

Using table lookup, the stem of given word can be found. Such a technique suffers from two major challenges: incompleteness and language dependency. All the possible terms of that language might not be present in the stem table, which results to incompleteness. Indexing over search tables makes the search faster, but as the size of stem table increases, B-trees or Hash tables are required. Hafer and Weiss[18] gave the successor variety technique in which first of all, the successor variety of given term is identified. For example successor variety of given term "basket" in the available list 'bench.bake.bubble' will be 'e','a' and 'u', thus there are three successor varieties of "basket". As only 'a' stands as correct match, word 'bake' will be opted and same procedure will be carried out. Once the successor varieties are found, the information gathered during this process is used for creating segments in the word. Then after which one of the segments will be identified as stem. Before this technique is used, preparation of suffix lists and suffix removal rule set is required. N-gram stemmer uses statistical measures to identify the boundary of morphemes based on the distribution of morphemes in already seen set (training set). Also stemming algorithms based on Hidden Markov Model can be found in [24][25]. In such stemmers, the transitions can be defined by probability functions and transition between states defines the word building process. The path with highest probability from initial to final states will produce the split point from where the stem and its suffix can separate. Thus such stemmers don't require any prior linguistic knowledge.

3. STEMMING TECHNIQUES IN INDIAN LANGUAGES
Yet Another Suffix Stripper(YASS) was proposed by Majumdar et. al.[12] that is based on clustering technique and string distance concept. According to them, stemming improves recall of IR systems for Indian languages like Bengali. Dasgupta and Ng[10]worked on unsupervised morphological parsing for Bengali text and their technique analyzed the text without having prior knowledge of the language morphology. They evaluated their algorithm on set of 4110 human segmented Bengali words and achieved F-Score of 83%. DasGupta and Khan[8] has used three components Lexicon, Morphotactics

and Orthographic Rules as suggested in [23] and added one more component "Word Grammar" which used FSA optimally and provided full feature of specification of word. Thus their work was feature based. Criteria to judge stemmers are correctness, retrieval effectiveness and compression performance [14]. According to Bharati et. al.[18], a stemmer is correct if the words that are morphologically variants of each other, conflate to a single stem and the words that conflate to the single stem must be morphological variants of each other. Apart of this, the stem may not be the real morphological root of that word. Overstemming happens when two or more words, which are not morphological variant conflate to a single stem. Ramanathan and Rao[4] created suffix list and performed the longest match stripping on Hindi Words. They evaluated their stemmer using several Hindi news magazines containing 35977 unique words. The overstemming and understemming error percentages were 13.84 and 4.68 respectively. Their stemmer handled inflectional morphology by large, and did not accounted for most of the derivational morphology. This work was compared by Pandey and Siddiqui[6], who used GoldSmith's approach for stemming Hindi text. Their unsupervised learning technique resulted to 89.9% accuracy with F-Score 94.96%.

4. GUJARATI STEMMER AND PROPOSED WORK

In Gujarati language, there are three genders namely masculine, feminine and neuter, two numbers that are singular and plural, three persons (first, second and third) and three cases of nouns namely nominative, vocative and locative. Gujarati is free order language, unlike English, the nouns, verbs and adjectives are highly inflectional by nature. For Gujarati language, the stemmers proposed till now are of hybrid nature. Thus, apart of handcrafted rules, the researchers have also used unsupervised technique as proposed in [14], through which optimal split position is found. The accuracy of such techniques is found 67.86% [1]. The approach for derivational stemming as given by Jurafsky and Martin[23]was used by Kartik Suba and Pushpak Bhattarcharya which resulted to 70.7% accuracy for Gujarati language[1]. They also experimented an inflectional stemmer by using 73 hand-crafted rules, using POS tagging module and Goldsmith's take all splits method[14] which gave 90.7% accuracy[1].

4.1. Model Proposed

The stemmers defined for Gujarati till now, have not been evaluated in terms of overstemming and understemming. Also, the incorporation of Named Entity Recognition is missing. The Gujarati Stemmer proposed by the authors is described in Figure 1 as follows:

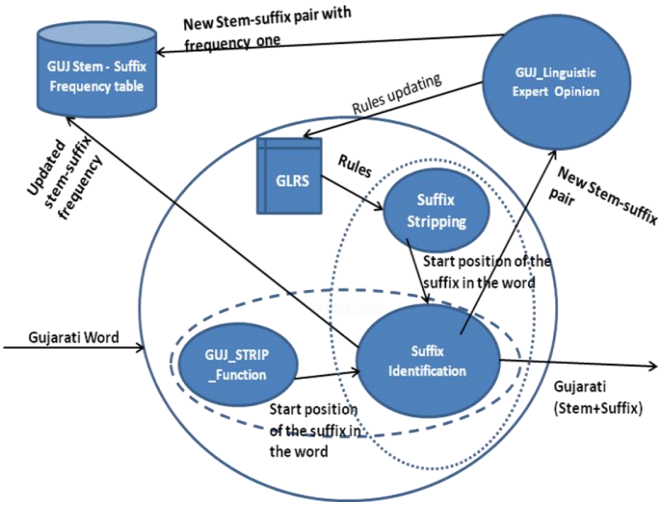

Fig 1. Proposed GUJARATI_STEMMER

4.1.1. Components of the Stemmer

GUJ Stem-suffix frequency table (GSFT) stores the frequency of stem-suffix pair based on the training done on a Gujarati Corpus. This is helpful to determine the various inflections possible with a stem.

Gujarati Language Rule Set(GLRS) stores the rules for suffix stripping. These rules are defined based on the study of the characteristics of Gujarati language. For example, word કર(to do) has suffixes like વું, તાં, તી, etc. Same suffixes are also valid with word પડ(to fall). Thus one can derive a rule to strip off such suffixes and even many more of such kind.

The rules of GLRS are used by suffix stripping module so as to find the start position of the suffix in the given word and thus separate the stem and the suffix. This would also update the frequency of the stem-suffix pair in GUJ Stem-suffix frequency table (GSFT).The authors propose a GUJ_STRIP_function which is based on artificial neural network technique alongwith the probability and smoothing concept. If the rule set is not able to help in the identification of the stem, this function will help to deduce it. For this, the function uses the GUJ Stem-suffix frequency table to find probability of the available suffixes known till now. Apart of this, it uses smoothing techniques to determine the probability of a suffix which has not been seen till now. To identify words which are names of people or places, named entity recognition techniques are to be incorporated in the model.

Using this function, if some new kind of suffix is identified then that stem-suffix pair is sent to the Gujarati linguistic expert for validation and once approved, this would get added to the GUJ Stem-suffix frequency table.

4.1.2. Suggestions with respect to Gujarati Stemmer development:

As Gujarati is highly inflection in nature, overstemming and understemming may happen. For example, વહેતું(flowing) andવહેમ (superstition) may result to same stem વહે(to flow), but they are not morphological variants. Understemming implies that though both the words are morphogical variant, they don't conform to same stem. E.g. ખવડાવવું(to make someone eat)

and ખાવું(to eat) are morphological variants of ખા(eat), but during the stemming, the previous one may conflate to ખવડાવ and the latter one to ખા due to poor rule set or stemming technique. The retrieval effectiveness may be measured based on precision and recall measures [14]. The authors suggest that currently the work done on stemming for Gujarati language is measured in terms of correctness but not in terms of incorrectness i.e. overstemming and understemming. Apart of evaluation of stemming also if Named Entity recognition is used, ontology gets incorporated and Fuzzy techniques are applied, the stemmers would become more efficient.

It was found that POS tagged data improved the performance of English stemmer [20]. The same was found for the Hindi stemmer [19]. Thus we suggest that building a POS tagger for Gujarati language will also improve Gujarati stemming as both the languages belong to Devnagari scripts and share several common features.

5. CONCLUSION

The authors have proposed a model for Gujarati Stemmer which not just focuses on traditional approach but also incorporates naïve approaches to derive an accurate and efficient stemmer. As the research in Gujarati language is much less than those in languages like English and Hindi, still lot of dimensions are untouched. Using several relevant and new trends, better Gujarati stemmer can be developed and thus will be useful for further linguistic computing.

6. REFERENCES

[1] Patel P., Popat K. and Bhattacharyya P. 2010. "Hybrid Stemmer for Gujarati", Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), The 23rd International Conference on Computational Linguistics (COLING), pp. 51–55.

[2] Suba K., Jiandani D. and Bhattacharyya P. 2011. "Hybrid Inflectional Stemmer and Rule-based Derivational Stemmer for Gujarati", Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011, pp. 1–8.

[3] Larkey L., Connell M. and Abduljaleel N. 2003. "Hindi CLIR in Thirty Days", ACM Transaction on Asian Language Information Processing, Vol-2(2), pp. 130-142.

[4] Ramanathan A. and Rao D., 2003. "A Lightweight Stemmer for Hindi", Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, pp. 43-48.

[5] Shrivastava M., Agrawal N., Mohapatra B., Singh S. and Bhattacharya P. 2005. "Morphology Based Natural Language Processing Tools for Indian Languages", Proceedings of the 4th Annual Inter Research Institute Student Seminar in Computer Science, IIT, Kanpur, India.

[6] Pandey A. and Siddiqui T. 2008. "An Unsupervised Hindi stemmer with heuristic improvements", Proceedings of the second workshop on Analytics for noisy unstructured text data (AND'08).

[7] Dolamic L. 2010. "Comparative Study of Indexing and Search Strategies for the Hindi, Marathi and Bengali Language", ACM Transactions on Asian Language Information Processing (TALIP), Vol. 9(3).

[8] Dasgupta S. and Khan M. 2004. "Feature unification for morphological parsing in Bangla", Proceedings of the 7th International Conference on Computer and Information Technology.

[9] Bhattacharya S., Choudhury M., Sarkar S. and Basu A. 2005. "Inflectional Morphology Synthesis for Bengali Noun, Pronoun and Verb Systems", Proceedings of the National Conference on Computer Processing of Bangla (NCCPB 05), pp. 34-43.

[10] Dasgupta S. and Ng V. 2007. "Unsupervised Morphological Parsing for Bengali", Language Resources and Evaluation, DOI 10.1007/s10579-007-9031-y, Springer Science + Business Media B.V.

[11] Islam M., Uddin M. and Khan M. 2007. "A Light Weight Stemmer for Bengali and Its Use in Spelling Checker", Proceedings of the 1st International Conference on Digital Communications and Computer Applications (DCCA2007).

[12] Majumder P., Mitra M., Parui S. and Kole G. 2007. "YASS: Yet another suffix stripper", ACM Transactions on Information Systems (TOIS).

[13] Das A. and Bandyopadhyay S. 2010. "Morphological Stemming Cluster Identification for Bangla", Knowledge Sharing Event-1: Task 3: Morphological Analyzers and Generators.

[14] Frakes W. and Baeza-Yates R. 1992. Information Retrieval: Data Structures and Algorithms. Prentice Hall, Englewood Cliffs, New Jersey, USA.

[15] Goldsmith J. 2001. "Unsupervised learning of the morphology of a natural language", Computational Linguistics, Vol. 27(2), pp.153-198.

[16] Porter M. 1980. "An algorithm for suffix stripping". Program, Vol. 14 (3) pp. 130-137.

[17] Paice C. 1974. "Another Stemmer", ACM SIGIR forum, Vol. 25(3), pp 56-61.

[18] Bharati A., Chaitanya V., and Sangal R. 1995. Natural Language Processing: A Paninian Perspective. Prentice Hall of India, New Delhi, India.

[19] Hafer M. and Weiss S. 1974. "Word Segmentation by letter successor varieties", Information Storage and Retrieval, Vol. 10(1), pp. 371-385.

[20] Shrivastava M. and Bhattacharyya P. 2008. "Hindi POS Tagger Using Naive Stemming: Harnessing Morphological Information Without Extensive Linguistic Knowledge", Proceedings of International Conference on NLP (ICON08).

[21] Minnen G., Carroll J., and Pearce D. 2001. "Applied morphological processing of English", Natural Language Engineering, pp. 207-223, Cambridge University Press.

[22] Lovins J. 1968. "Development of a Stemming Algorithm," Mechanical Translation and Computational Linguistics, Vol. 11(1) and Vol. 11(2), March and June 1968.

[23] Jurafsky D. and Martin J. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall, Englewood Cliffs, NJ.

[24] Massimo M. and Nicola O.. 2003. "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.

[25] Alajmi A., Saad E., and Awadalla M. 2011. "Hidden markov model based Arabic morphological analyzer", International Journal of Computer Engineering Research Vol. 2(2), pp. 28-33.