Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 6, June -2015

A SURVEY OF DIFFERENT STEMMING ALGORITHM

Dhaval Patel[1], Prof.Miral Patel[2], Prof.Yogesh Dangar[3]
[1,2,3]IT Department, GCET, V V Nagar, India

ABSTRACT: *Stemming is the process of knowing the root word from derived form. It is one of the important operation in context of Information Retrieval system for search query analysis and Machine Translation System. Stemming can be achieved by removing affix from the transformed form of base one. For designing perspective, it is based on linguistic knowledge in form of rules or statistical knowledge gained from huge collection of monolingual words. This paper covers survey of different stemming approaches, both ruled based as well as statistical based.*

Keywords: *Stemming; Rule based; Statistical ; Survey*

1. INTRODUCTION

Natural Language Processing(NLP) is a field which mainly devoted to make computer to learn human language and to process it intelligently. To understand a language, analysis has to be done at word level, sentence level, context level and discourse level. Morphological analysis comes in the base of all, as it is first step to understand a given sentence. One of the tasks that can be done at morphological level is stemming.

For any language, mostly the root words are not used in exact form to convey a message. They are always transformed by attaching affix to convey exact message, i.e. help, helpful, helpless.

It has been observed that most of the times the morphological variants of a word have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. Since a meaning is same but a word form is different, it is necessary to identify each word form with its base form.

This process of retrieving root word from conflation is known as Stemming. Stemming can be achieved by removing affix, either prefix or suffix from a word.

For any machine translation system one of the important thing is to know the morphology, and stemming is preprocessing tool used for that.

2. APPROACH FOR DEVELOPING STEMMER

Stemming algorithms have been developed to convert the morphological variants of a word like "introduction", "introducing", "introduces" etc. to get mapped to the word "introduce". It is important to note that stemming does not mean to retrieve root form which is grammatically correct or available in dictionary.

Some algorithms may map above words to just "introduc", but that is allowed as long as all of them map to the same word form, more popularly known as a stem form. Stemming usually refers to a crude heuristic process that chops off the end of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. For example, the word inflations like "gone", "goes", "going" will map to a stem "go", a word "went" will not map to the same stem.

In stemming process, the errors are occurred because of over-stemming and under-stemming. Under-stemming is occurred when words that refer to the same concept are not reduced to the same stem. This will cause a failure in conflating related words[4][11]. An example of understemming in English, would be: "compile" being stemmed to "comp", and "compiling" to "compil".

Over-stemming occurs when words are converted to the same stem even though they refer to distinct concepts [4]. It occurs in case of conflation of semantically distant words. This can result in the conflation of unrelated words. For instance, "compile" and "compute" getting stemmed to "comp". Out of them, Over-stemming cause misleading to user by providing irrelevant result and hence need to overcome by proposed algorithm.

There are number of algorithms have been proposed for developing stemmer, Based on the nature of algorithm they are mainly divided into two approaches

- Rule based Approach
- Statistical Approach

3. RULE BASED APPROACH

This category of algorithm is purely based on the morphological knowledge of language. In this approach the transformation rules are designed, which may be in either substitutional form or affix removal form based on the language and then they are applied. The following algorithms are of this kind.

3.1 Lovins Algorithm

It is the first stemmer proposed by Lovins in 1968. It performs a lookup on a table of 294 endings, 29 conditions and 35 transformation rules and it is set based on a longest match principle[1].

The algorithm first eliminate a longest suffix from a word and then word is processed with various adjustments mainly to retrieve valid word[1].

Lovins stemmer is a single pass algorithm. So it always removes maximum of one suffix from a word[1].

3.2 Porter Algorithm

Porter (1980) proposed an algorithm for suffix stripping. It is rule based and is used for English like language which is less inflectional. The rules are expressed in the form of,

(condition) S1 -> S2 [3]

It is depicted from above rule that if a word having suffix S1, then it is replaced by S2 with the given condition are satisfied by. Here condition usually stated in form of length of the stem (m). For example[3],

(m > 1) EMENT ->

Here it is applied for i.e. REPLACEMENT to REPLAC. This process does not eliminate suffix for the short stem where the stem length being specifies by m. There is no linguistic basis for this approach[3].

3.3 Paice Algorithm

Paice (1990)[4] proposed another stemmer based on Table containing around 120 rules. It is an iterative algorithm and arrangement of rules in table based on the last letter of a suffix. Now each repeating time, the rule is selected based on last character of the word. Each rule specifies either a deletion or replacement of an ending[4].

It terminate if no appropriate rule for last character of word or a word having only two letters with vowel as initial or a word having only three letters with consonant as initial[4].

3.4 Advantage of Rule Based Approach:

- It is much faster because it does not requires any preprocessing steps.
- It is purely based on Language under consideration so more reliable compare to the corpus based approaches, as statistical training purely based on Corpus provided.

3.5 Disadvantage of Rule Based Approach:

- It requires extensive knowledge of language in order to form rules.
- For better stemmer, it requires rules to cover all morphologic form of language. Otherwise it produces undesirable result for uncover morphology.
- Sometime it is hard to formulate rules. For example, in English some past particle forms are different than conventional having suffix "ed". Exceptions thus need to be formulated for a set of words.

4. STATISTICAL APPROACH

This approach is independent of Language as knowledge of the morphology of language is gained by statistical approach. So no need to form rules initially as a case with Rule based Approach. In this approach statistical knowledge is obtained by well formed Corpus. For this approach, two sets of Corpus is formed, one for training purpose and second one used for testing purpose to check accuracy.

4.1 Corpus Based Approach

Xu and Croft (1998) proposed corpus based approach for errors found in Stemming process. [5].

The basic idea is to generate equivalence classes for words with a classical stemmer and then separate some conflated words based on their co-occurrence in the corpora[5]. Because of it, the algorithm avoids incorrect conflations such as "policy/police". Thus it overcome Porter's Error[5].

4.2 Goldsmith Approach

Goldsmith (2001) proposed an algorithm for Unsupervised learning of Morphology based on heuristic and information theory[6]. The proposed process learns stem of words based on the minimum description length (MDL) [6].

A initial heuristic is used to define a probabilistic conflation of word and employ MDL in order to decide whether proposed probabilistic work are accepted or not. Apart from initial, other incremental heuristics are used to improve the results.

For experiment purpose, the Gold standard corpus is designed which has around 15000 words The results matches well with analysis that expressed by a Language Expert[6].

4.3 N gram Approach

Mayfield and McNamee (2003) proposed single N-gram stemming algorithm. It is purely based on choosing a single N gram as stem for a word. It can be an effective and efficient language-neutral approach for some languages[7].

The design of approach is to explore distribution of all N grams in a document. This approach is based on the phenomena that suffix are more frequently occurring part in document than stem. The idea is some of the N grams extracted from word will cover only portions of the word that do not show morphological variation[7].

For example, the words "juggle", "juggling" and "jugglers" share the common 5-gram "juggl". And to identify the relationship between N gram, inverse document frequency (IDF) is used [7].

4.4 HMM based Approach

Massimo and Nicola (2003) proposed a novel statistical method for stemmer generation based on Hidden Markov models[8]. It is based on unsupervised leaning with no prior knowledge or manually created training set.

HMMs are finite-state automata with transitions defined by probability functions. Each character in a word is treated as a state[8].

A HMM topology defines the number of states, the labeling of states as belonging to one of the two sets, the allowable initial and final states, and the allowable transitions. Yet all the probability functions that constitute the HMM parameters need to be computed.

The transition is controlled by Probability function. At each transition, new state emits a Symbol and associated probability. For any word, the optimal path from initial state to final state provide a split of word. And character set before split point is considered as Stem while rest as suffix. The authors considered three different topologies of HMM in their experiments[8]. Using Porter's algorithm as a baseline, they found that HMM had a tendency to over stem words[8].

4.5 Yet Another Suffix Stripper

Majumder et al. (2007) developed statistical approach YASS: Yet Another Suffix Stripper. It is based on clustering technique formed using string distance measures and requires no linguistic knowledge[9].

A set of string distance measures {D1,D2,D3,D4} are defined and used for clustering the words. Here the distance function maps a pair of string a and b to real number r, where a smaller value of r indicate greater similarity between a and b.

Given two strings X = x0x1 . . . xn and Y = y0 y1 . . . ym, we first define a Boolean function pi (for penalty) as follows:

pi = 0 if xi = yi 0 = i = min(n, m)
1 otherwise

Thus, pi is 1 only if inequality found at the ith position of X and Y. The distance functions pointed out above are used to cluster words into homogeneous groups. Each group is expected to represent an equivalence class consisting of morphological variants of a single root word [9]

4.6 Advantage of Statistical Approach:

- Mainly it does not need any linguistic expertise. So it can be used for Language that are not more explored (The resources are not available).
- It is best suited for Language that primarily “suffixing” in nature.

4.7 Disadvantage of Statistical Approach:

- It is more Time consuming than Rule based as it need to perform preprocessing task.
- Corpus size is important, as corpus size decreases, the possibility of covering most morphological variants will also decrease, resulting in a stemmer with poorer coverage.
- It produce wrong stem for a word where only suffix removal is not sufficient but also require some substitution. For ex. "Loving" to "Love + ing " not " Lov + ing "[6].
- In some cases, it is not clear what the right form for the suffix is. For ex. "churches" to be "church" plus "s" or plus "es" [6].

5. CONCLUSION

The stemming is one approach used in indexing process. This paper covers mainly both approaches for developing stemmer. The rule based approach is used for developing aggressive stemmer, when having linguistic support available for language. But it will produce both over stemming and under stemming error.

The statistical approach is best suited for language that are not explored widely and having primarily suffixing nature. For this approach, there is no need of Language Expertise but they need Corpus through which statistical is acquired. So it is required to have corpus that cover all possible morphological variants. In order to improve the result of stemmer, combination of both approach means Hybrid approach is used which overcome pitfalls of both of above[19].

Table 1. Analysis of Different Stemming Algorithms

Stemming Algorithm	Advantage	Disadvantage	Applied Language	Analysis
Porter Stemmer	• It is light weight stemmer than Lovins[4].	• It is time consuming because of five steps process.	Portuguese	Observed Understemming and Overstemming Error
			English	UI is closed to Lovin but less OI than other stemmer[4]

Paice Stemmer	• It is simpler. • Rules are designed for both removal and replacement.	• It is Heavy weight stemmer. • Have higher Overstemming than others [4]	Portuguese	Observed Understemming and Overstemming Error
			English	Less UI than other
Goldsmith Approach	• Does not need any language knowledge. • Used for More Morphologically rich language	• Mainly dependent on cleanliness of Corpus. • Execution time is higher for larger corpus at initial stage.	Telugu	F-score around 92%
			Hindi	F score around 94% [10]
N-Gram Approach	• Does not need any language knowledge.	• Performance penalty is obvious. • Disk usage is High	Marathi	Accuracy around 82.5%
HMM based Stemmer	• Does not need any language knowledge	• Overstemmed a word	English	Retrieval effictiveness is same as Porter [8]
YASS	• Use statistical Approach. • Best suited for Language that are primarily suffixing in nature	• Corpus size does matter for covering variants. • Handling excessive conflation	Bengali	Achieved F score around 83% [9]

6. REFERENCES

[1] Lovins, J.B. Development of a stemming algorithm. Mechanical translation and Computational Linguistics 1968; 11, 22-31.
[2] D. John, "Suffix removal and word conflation", ALLC Bulletin, Volume 2, No. 3, 33-46, 1974.
[3] M. Porter, "An Algorithm for Suffix Stripping Program", 14(3): 130-137, 1980.
[4] C. D. Paice, "Another stemmer". ACM SIGIR Forum, Volume 24, No. 3, 56-61, 1990.
[5] X. Jinxi and C. Bruce W., "Corpus-based Stemming Using Co-occurrence of Word Variants", ACM Transactions on Information Systems, Volume 16, Issue 1, 61-81, 1998.
[6] J. A. Goldsmith, "Unsupervised Learning of theMorphology of a Natural Language", Computational Linguistics, MIT Press, 27(2):153-198, 2001.
[7] J. Mayfield and P. McNamee, "Single N-gram stemming", Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 415-416, 2003.
[8] M. Massimo and O. Nicola. "A Novel Method for Stemmer Generation based on Hidden Markov Models", Proceedings of the twelfth international conference on Information and knowledge management, 131-138, 2003.
[9] P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra, and K. Datta, "YASS: Yet Another Suffix Stripper", Association for Computing Machinery Transactions on Information Systems, 25(4):18-38, 2007.
[10] A. K. Pandey and T. J. Siddiqui, "An Unsupervised Hindi Stemmer with Heuristic Improvements", In Proceedings of the Second Workshop on Analytics For Noisy Unstructured Text Data, 303:99-105, 2008.
[11] Viviane Moreira Orengo, Christian Huyck "A Stemming Algorithm for the Portuguese Language", String Processing and Information Retrieval,13-15 Nov. 2001
[12] F. Peng, N. Ahmed, X. Li and Y. Lu, "Context Sensitive Stemming for Web Search", Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 639-646, 2007.
[13] M. Creutz and K. Lagus, "Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora using Morfessor 1.0.", Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology, 2005.
[14] A. Ramanathan and D. D. Rao, "A Lightweight Stemmer for Hindi", Workshop on Computational Linguistics for South-Asian Languages, EACL, 2003.
[15] M. Z. Islam, M. N. Uddin and M. Khan, "A Light Weight Stemmer for Bengali and its Use in Spelling Checker". Proc. 1st Intl. Conf. on Digital Comm. and Computer Applications (DCCA 07), Irbid, Jordan, March 19-23 2007.
[16] S. Dasgupta and V. Ng, "Unsupervised Morphological Parsing of Bengali", Language Resources and Evaluation, 40(3-4):311-330, 2006.

[17] M. M. Majgaonker and T. J Siddiqui, "Discovering Suffixes: A Case Study for Marathi Language", International Journal on Computer Science and Engineering, Vol. 02, No. 08, pp. 2716-2720, 2010.
[18] K. Suba, D. Jiandani and P. Bhattacharyya, "Hybrid Inflectional Stemmer and Rule-based Derivational Stemmer for Gujarati", In proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011, Chiang Mai, Thailand, pp.1-8, 2011.
[19] Ms. Jikitsha Sheth, Dr. Bankim Patel "Dhiya: A Stemmer for morphological level analysis of Gujarati language", IEEE, International Conference on 7-8 Feb. 2014

