ISM@FIRE-2013 Morpheme Extraction Task

Amit Jain, Nitish Gupta, Sukomal Pal
Dept. of CSE,
Indian School of Mines, Dhanbad, India.
(amitjain.bagra@gmail.com , nitish.gupta183@gmail.com , sukomalpal@gmail.com)

Abstract

This is third year of participation from Indian School of Mines Dhanbad at FIRE. At FIRE-2013, we have participated in Morpheme Extraction Task (MET).In MET,we submitted a morpheme extraction tool which takes input a single file of documents in any language and produces a two-column file listing words in the language and their corresponding stemmed root words.

Introduction

Our objective for participation is to compare the efficiency and performance of ISMStemmer in comparison to other stemmers in several Indian languages. ISMstemmer, being a language-independent stemmer, worked on a set of Indian languages and performed better than baseline retrieval results for Marathi, Gujarati, Bengali and comparable to baseline for Odia language.

The paper is organized as follows. Section 2 discusses about related works. Section 3 focusses on stemming. Section 4 discusses about our participation and results and section 5 comprises of conclusion.

2 Related work

Stemming is a popular technique in IR which has proven to enhance recall in general, but for the morphologically complex languages improves precision as well [2]. Stemming algorithms are broadly classified into two categories, namely rule-based [3] and statistical [4, 5, 1]. Rule-based stemmers work on a set of pre-defined language-specic rules, whereas statistical stemmers employs statistical information from a large corpus of a given language in order to learn the morphology. A statistical stemmer obviates the need of language-specific expertise, and therefore, is often a preferred choice, specifically in information retrieval [2,4, 1]. There are mainly three kinds of approaches to language-independent stemming. The first kind of methods take a set of words and try to find probable stems and suffixes for each word; other methods look for association between lexicographically-similar words by analyzing their co-occurrence in a corpus. The third group of methods is based on character n-grams.
Paik et al. [6] proposed a graph-based statistical stemmer GRAS, where a set of word classes are formed each having a pivot word or stem. All the words within a class share a common prefix but have different valid suffixes. Valid suffixes are shortlisted pairwise based on their occurrence with other prefix words in the corpus.

Majumder et al. [1] developed a clustering-based unsupervised technique (YASS) where string-distance between two words is used. A long-match in the prefix for a word pair is rewarded while an early mismatch is penalized during the complete-linkage clustering. Bacchin et al. [5] described a probabilistic model for stem generation based on the mutual reinforcement relationship between stems and suffixes. Stems and suffixes are generated by splitting words at all possible positions. A set good stems and good suffixes are chosen using HITS algorithm. In the work by Oard et al. [4], suffixes were discovered statistically in a text collection and the word endings were eliminated. The frequency of every one, two, three, and four character suffix that would result in a stem of three or more characters for the first 500,000 words of the collection were collected. Then they subtracted the frequency of the most common subsuming suffix of the next longer length from each suffix (for example, frequency of "ing" from the frequency of "ng"). The adjusted frequencies were then used to sort all n-gram suffixes in descending order. The count vs rank plot was found to be convex and the rank where minimum frequency was reached was chosen as the cutoff limit for the number of suffixes for each length. Our approach is to some extent close to this work. However it differs in the fact that we try to discover suffixes based on the concept of frequent itemset generation technique according to apriori algorithm in market basket data analysis [7].

3 Stemming

The usefulness of stemming is shown to be mixed in languages such as English. But in case of morphologically complex languages, stemming produces a significant performance improvement. A number of linguistic rule-based stemmers are available for most European languages which employ a set of rules to get back the root word from its variants. But for Indian languages which are highly inflectional in nature, devising a linguistic rule-based stemmer needs some additional resources which are not available. We present a purely statistical and corpus based approach which extracts morphemes from inflected form of words.

For morpheme extraction we have used a stemmer developed here in ISM [1] which strips off suffixes from inflected words. This is language-independent as we do not use any language-specific grammar rule. It works on the frequency of words. Stemmer extracts the morphemes from the words depending on their frequency of occurrence (of suffixes). We first find the list of frequent suffixes and then we extract the morphemes by removing the suffixes from the inflected word form.

We first identify in the lexicon a set of all suffixes of length n (n = 1, 2 . . .) by grouping words that share the same suffix and the number of words in a group becomes the frequency of the corresponding suffix. A suffix is called a potential suffix if its frequency in the lexicon is larger than a certain cut-off threshold. The rationale is that variant word forms of a language are generated by adding suffixes taken from a finite set of suffixes specific to the language and given a large enough corpus they also occur sufficiently frequently. So the frequency is a good indicator of the potentiality of a suffix.

In order to find the valid suffixes reverse the strings appearing in the single column file (list of unique unstemmed words) . Than perform the lexicographical sorting on these words so that all

the words sharing the common suffixes will be together. After that derive the suffixes from the entire corpus depending on the predefined threshold value. Only the characters having frequency higher than the threshold qualify as 1-character suffix. The set of 1-character suffixes also serve as the potential candidates while generating 2-character suffixes. A 1-character suffix which did not cross the threshold cannot be part of a 2-character suffix since its frequency is expected to be less or equal to that of 1-character suffix. Hence discard such 1-character suffixes during discovery of 2-character suffixes. Scan again and collected frequency for all possible 2-character suffixes. The set of 2-character suffixes crossing the threshold are considered valid 2-character suffixes and form the candidate-set for 3-character suffixes. Continue the process as long as a frequent n character suffix can be generated i.e. if the frequencies of such suffixes are greater than the predefined threshold value then those suffixes are valid suffixes.

The stemmer works in the following phases:

1. Find valid suffixes list
   a. Reverse the unique sorted word file.
   b. Again sort the reversed word file.
   c. Find the frequent suffixes (of length 1-character, 2-characters and so on).
   d. Find valid suffixes whose frequency is above a pre-decided threshold value α.

2. Reverse the original single column file.

3. Match each of the reversed word with the valid suffix list. If the word contains any of the suffixes, strip the matched suffix from the word such that the stemmed word has a longest prefix of length at least the threshold β.

4. Reverse the stemmed word to get the original stemmed word.

By the above mentioned algorithm we can find the stemmed word list which is used during indexing of document corpus.

4 Our Participation

- **Morpheme Extraction Task**

The language wise results for the Morpheme Extraction Task (MET), FIRE 2013 mentioned below.

IR Evaluation:

Table: Shows the Mean Average Precision(MAP) values obtained for different languages by the systems submitted. The MAP for Basline run (run with no stemming) is also given. Fourth column contains %improvement over the baseline score. The system from Indian School of Mines (ISM-Dhanbad) is language-independent and the system submitted by Maulana Azad national Institute of Technology (MANIT) is a morpheme extractor for Hindi. The hyperlinks provide query-wise full Trec result sets.

ISM

Language	Baseline	MAP Obtained	% improvement
Bengali	0.2740	0.3158	15.25%
Hindi	0.2821	0.2793	-0.99%
Gujarati	0.2677	0.2824	5.49%
Marathi	0.2320	0.2797	20.56%
Odia	0.1537	0.1583	2.99%

MANIT

Language	Baseline	MAP Obtained	% improvement
Hindi	0.2821	0.2917	3.40%

Linguistic Evaluation:

Table: Linguistic Evaluation is available for Tamil and Bengali. Anna University has submitted system for Tamil. The ISM's language independent system has been evaluated for Tamil and Bengali only for non-affixes as affix information is not given by the system. The hyperlinks provide full information on the five random samples that have been used for evaluation. Sample of 1000 words have been used from the gold standard test set to extract pairs of words having common morphemes for comparison.

AUKBC

Precision: 17.08%; non-affixes: 75.19%; affixes: 2.50%
Recall: 85.09%; non-affixes: 91.46%; affixes: 83.19%
F-measure: 28.45%; non-affixes: 82.53%; affixes: 4.85%

ISM-2013

Tamil:

Precision: 80.22%; non-affixes: 80.22%
Recall: 18.86%; non-affixes: 18.86%
F-measure: 30.54%; non-affixes: 30.54%

Bengali:

Precision: 60.64%; non-affixes: 60.64%
Recall: 32.15%; non-affixes: 32.15%
F-measure: 42.02%; non-affixes: 42.02%

5. Conclusion

This paper describes our participation in FIRE-2013 Morpheme Extraction Task (MET). We have used ISM Stemmer developed at ISM.

After participating in Morpheme Extraction Task (MET) we can say that our Morpheme Extraction tool is working well on all languages on which it has been tested out. Although the performance on Hindi Language is below reference level. On the basis of above reasons we can say that our Morpheme Extraction tool is an efficient multilingual tool. We would like to improve its performance on Hindi language as well in future.

6. References:-

[1] Raktim Banerjee, Sukomal Pal- ISM@FIRE-2011 Bengali Monolingual Task: A frequency based stemmer, Post-proceedings of FIRE-2011, IIT Bombay. (*accepted*).

[2] www.**isi**cal.ac.in/~**fire**/

[3] Cristopher D.Manning, Prabhakar Raghawan, Hinrich Schutze- An introduction to information retrieval, Cambridge University press 2008.

[4] http://en.wikipedia.org/wiki/Information_retrieval

[5]] https://sourceforge.net/p/lemur/wiki/Indri%20Query%20Language%20Reference/

[6] www.Lemurproject.org.

[7] Paik, J. H., Mitra, M., Parui, S. K., and J¨ arvelin, K. 2011. GRAS: An effective and efficient stemming algorithm for information retrieval. *ACM Trans. Inf. Syst. 29, 4, Article 19 (November 2011),* 24 pages.

[8] Paik, J. H. and Parui, S. K. 2011. A fast corpus-based stemmer. ACM Trans. Asian Lang. *Inform. Process. 10, 2, Article 8 (June 2011),* 16 pages.

[9] Paik J. H.**,** Pal Dipasree**,** Parui S. K. A Novel Corpus-Based Stemming Algorithm using Co-occurrence Statistics. *SIGIR'11,* July 24–28, 2011, Beijing, China.

[10] Xu, J. and Croft, W. B. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst. 16,* 1, 61–81.

[11] http://en.wikipedia.org/wiki/Stemmer

[12]. How Effective Is Suffixing? Donna Harman. *lister Hill Center for Biomedical Communications, National Library of Medicine, Bethesda, MD 20209*