A Simple Stemmer for Inflectional Languages

Jiaul H. Paik
CVPR Unit
Indian Statistical Institute, Kolkata
jia_r@isical.ac.in

Swapan K. Parui
CVPR Unit
Indian Statistical Institute, Kolkata
swapan@isical.ac.in

ABSTRACT

Stemming, a well known IR module, is used to enhance the effectiveness of text Retrieval Systems. In FIRE 2008 ad-hoc monolingual task we applied a simple corpus based technique that is based on n-gram matching on three Indian languages. In our method we group a class of words which share a common prefix of given character length and replace each of them by their common prefix. We hope to discover whether this simple method works well in Indian language context and the initial results are encouraging.

Keywords

Stemming, Information Retrieval

1. INTRODUCTION

Stemming refers to the process of transforming variant word forms to their root form. In Information Retrieval this process improves the ability of a system to match query and document vocabulary. It is similar to the morphological analysis in NLP, but is used to achieve somewhat differnt goals. For morphologically rich languages its purpose is twofold. On one hand it enhances recall. On the other hand it reduces the index size significantly (particularly for highly inflectional languages), thereby facilitating the keyword search process.

Almost all Indic languages are highly inflectional (except possibly Hindi). For example, Bengali can have more than 20 variants [3] for a single term. Inflection may occur mainly due to the addition of suffix/affix to terms, sometimes two separate terms combine to form a new term. Most existing stemmers, like porter [2] use a set of liguistic rules to fulfil the task. But for the languages, which have so far drawn little attention, building such stemmer requires some time. Even we sometimes can argue that such rule based stemmers performs as good as blind stemmers, particularly in the context of IR. The objective of our participation in FIRE 2008 was to experiment a simple corpus based unsupervised stemmer on several Indian languages. We tested it on three major Indian languages (Bengali, Hindi, Marathi) which are primarily suffixing in nature. The approach we took is entirely blind and corpus specific in that it needs no linguistic information.

The rest of this paper is organised as follows: Section 2 gives a brief overview of the test collections used in our experiment. Section 3 describes the adopted stemming technique. The term weighting formula and query-document matching technique are given in Section 4. Results are given in Section 5. In the concluding section we describe some of our observations about the data sets and some outlines of the problems that need to be addressed.

Table 1: Statistics on FIRE 2008 Test Corpora

Corpus →	BENGALI	HINDI	MARATHI
No. of queries	50	50	50
Raw text size(in MB)	738.6	690.6	491.3
No. of documents	123047	95215	99362
No. of words	540232	209546	854324
No. of rel. documents	1863	3436	1095

2. TEST COLLECTIONS

The corpora used in our experiments are newspaper articles collected over a period of time. Each document has only one logical segment, namely **text**. Each topic has three logical segments (title, description and narrative). Through eye inspection we noticed some serious errors in Hindi corpus (although both Marathi and Bengali corpus contain the same kind of errors that can do only limited damage), like consecutive occurrences of independent vowels, non-permissibile sequences of characters etc. We indexed the collection after cleaning such errors. Detailed statistics are given in the Table 1.

3. STEMMING PROCEDURE

Our algorithm computes the proximity of two terms with respect to the length of the common prefix. We now define the notion of a character in the context of Indian languages. In all Indian languages, two or more basic consonant characters sometimes combine together to form what is called a compound charcater. Also, vowels can occur in two different ways, first as an independent character and second as a dependent character. We now define a character as one of the following:
1. An independent basic consonant or vowel; 2. A basic consonant along with a succeeding dependent vowel. 3. A compound character with or without a suceeding dependent vowel.
The match length between two words is defined as the number of characters present in the common prefix. The central theme of our algorithm is to form an equivalence class of words such that all members of the class share a common prefix of length not less than a given length. The common prefix then becomes the root of the class. The algorithm

Table 2: Meaning of different symbols used

Symbols	Meaning
q	User's query
d	Document in the collection
t	Observed term
N	No. of documents in a collection
F	Total no. of tokens of an observed term t
n	Size of the elite set for term t
tf	Term frequency
qtf	Multiplicity of term-occurence in query
tfn	Normalized term frequency
avg_l	Average length of the document
$l(d)$	Length of the document d
$R(q,d)$	Relevance score of d given q

does not employ any expensive clustering algorithm like agglomerative or partitional method to detect the word clusters. It simply takes a sorted lexicon and determines all the clusters in a single linear scan. So on a sorted lexicon of size N the complexity of the algotithm is $O(N)$, which is $O(N^2)$ for an agglomerative clustering technique. Again in the clustering based approaches, parameter selection is not a trivial task. It varies from language to language and even sensitive to the nature of the corpus (for same language). In our approach we do it in a robust way, by considering only the matching prefix length.

4. IR MODEL USED

In all our experiments, we used TERRIER [4] for indexing and retrieval. All submitted results were retrieved using IFB2 model. This model is an approximation of Poisson and inverse document frequency mixture with two level normalization. In the first level, risk factor is considered in accepting a term as a document descriptor by measuring the ratio of two Bernoulli processes. The fundamental assumption behind the first normalization is that the gain is directly proportional to the amount of risk involved in a decision. The second level normalizes the term frequency by hypothesizing that the term frequency density is inversely related to the length of the document. We will show in result section why we chose it as our primary model. The query-document matching function is given by the following equation (1):

$$R(q,d) = \sum_{t \in q} qtf \cdot (1 - Prob_2(tfn)) \cdot Inf_1(tfn) \quad (1)$$

$$Prob_2(tfn) = \frac{F+1}{n \cdot (tfn+1)} \quad (2)$$

$$Inf_1(tfn) = tfn \cdot log_2 \frac{N+1}{F+0.5} \frac{F}{N} \quad (3)$$

$$tfn = tf \cdot log_2(1 + \frac{avg\_l}{l(d)}) \quad (4)$$

Higher the value of $R(q,d)$, higher is the degree of relevance of the document d with query q. The meaning of the symbols used in equations 1, 2, 3, 4 are given in Table 2.

5. EXPERIMENTAL RESULTS

Table 3: Training MAP for Bengali and Marathi

Corpus →	Benagli		Marathi	
Pref. len → Model ↓	3	4	3	4
BB2	.4843	.4664	.4273	.4663
BM25	.4580	.4343	.4727	.4434
DFR-BM25	.4779	.4564	.4683	.4688
DLH	.4449	.4507	.4737	.4666
IFB2	**.4916**	**.4672**	**.4745**	**.4910**
PL2	.4408	.4400	.4867	.4819
TF_IDF	.4586	.4360	.4684	.4601

Table 4: FIRE 2008 results

	Beng.1	Hindi.1	Hindi.2	Mara.1	Mara.2
MAP	.4232	.2558	.2709	.3862	.4239
R-prec	.4172	.2826	.2869	.3783	.4144
bpref	.3828	.2927	.3101	.3501	.3860
Num-Rel	1863	3436	3436	1095	1095
Rel-Ret	1784	2300	2462	1033	1030

5.1 Training

The purpose of training was twofold. First, we wanted to see which model works best for a particular language. Second, what should be a good prefix length for grouping the terms. For Bengali and Marathi we used FIRE 2008 document collection and topic 1-25 for training. We could not perform it on Hindi because of unavailability of relevance judgement data. In all the experiments primary evaluation measure was mean average precision. We noticed that the model **IFB2** performs consistently better than the other models in both the languages. When choosing prefix length threshold, we observed for Bengali 3 is a good choice and for Marathi it was 4. Table 3 shows the training results.

5.2 Official results

We submitted five runs: one for Bengali (Beng.1), two for Hindi (Hindi.1, Hindi.2) and two for Marathi (Mara.1, Mara.2). For all languages retrieval was performed using all given query fields (title, desc, narr). Bengali run was submitted with prefix length 3. Prefix length for Hindi.1 and Marathi.2 was 3 and that for Hindi.2 and Marathi.1 was 4. Table 4 gives the results for five official runs.

6. CONCLUSION

We tested a very simple approach which is very fast and also performs reasonably well. The performance of this stemmer is sensitive to the noise due to spelling mistakes, as it determines the rule from the corpus itself. Further investigation needs to be done particularly on the prefix length selection portion, as it may be language dependent. We hope this approach will be acceptable for all languages which are generally suffixing in nature.

Among the problems we noted, grouping of semantically unrelated terms into the same cluster is the most important. The approach taken by Xu and Croft [1] may be one solution, but a deeper semantic analysis is necessary to resolve proper noun vs abstract noun ambiguity, as because in Indian language context almost every proper nouns are abstract nouns. We hope to investigate these issues.

Acknowledgments
The authors would like to thank the FIRE organizers for their initiative and effort. Special thanks to Dipasree Pal for her help with TERRIER retrieval system.

7. REFERENCES

[1] Xu, J. and Croft, W.B., Corpus-Based Stemming using Co-occurrence of Word Variants. In ACM TOIS, volume 16(1), 1998.

[2] Porter, M. F. An algorithm for suffix stripping. Program 14, 3, 130–137, 1980.

[3] Majumder et al. Yass: Yet another suffix stripper. In ACM TOIS, volume 25(4), 2007.

[4] G. Amati and C. J. van Rijsbergen. Probabilistic models of Information Retrieval based on measuring the divergence from randomness. In ACM TOIS, volume 20(4), 2002.