N-gram Tokenization for Indian Language Text Retrieval

Paul McNamee
JHU Human Language Technology Center of Excellence
paul.mcnamee@jhuapl.edu

ABSTRACT

Character n-gram tokenization is a language-neutral technique that addresses the problems created by morphological processes that lower IR performance, such as inflection, derivation, and compounding. N-grams have been widely adopted for use in Asian languages, especially languages such as Chinese and Japanese where words are not separated by spaces. Use of n-grams in alphabetic languages is less popular; however, they have been shown to be an effective technique in many European languages using data sets developed at CLEF.

This paper describes monolingual experiments using n-grams as the primary method of tokenization in several Indian languages. Tests are conducted in Bengali, Hindi, and Marathi using benchmarks created in 2008 for the FIRE workshop.

Keywords

Multilingual text retrieval, Character n-grams, FIRE

1. INTRODUCTION

Character n-gram indexing works in every language, requires no training, and is more effective than raw words in morphologically richer languages. The redundancy provided by the method is useful for capturing root morphemes, in addition to other substrings, and thus is a means of controlling inflectional morphology. However the redundancy also increases the amount of disk storage required for an inverted file.

This paper reports experiments using test sets in four languages used in the FIRE 2008 workshop: Bengali, English, Hindi, and Marathi.

1.1 HAIRCUT

The Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT) information retrieval system was developed at the Johns Hopkins University Applied Physics Laboratory [3]. The software is written in Java and it supports modern IR techniques, including the language modeling retrieval framework [1, 6], which has become increasingly popular in recent years. HAIRCUT also supports n-gram tokenization, automated relevance feedback, and both dictionary and corpus-based translation.

No language-specific resources such as stopword lists, thesauri, or morphological analyzers were used in these experiments.

1.2 Language Models for IR

In the language model approach to retrieval documents are ranked for their relevance to queries based on a generative model. Specifically the probability that is being estimated is the maximum likelihood estimate that a relevant document, D, could be generated from a unigram language model based on the query, Q, is $P(D|Q)$. Because queries tend to be much shorter than documents it is very difficult to estimate this probability directly, therefore Bayesian inversion is applied:

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \tag{1}$$

If we make the assumption that *a priori* all documents are equally likely to be relevant regardless of Q we obtain:

$$P(D|Q) = \frac{P(Q|D)}{P(Q)} \tag{2}$$

For the purpose of ranking of documents in decreasing likelihood of relevance we can omit the prior probability of the query, Q, leaving:

$$P(D|Q) \propto P(Q|D) \tag{3}$$

Assuming that terms are independent:

$$P(D|Q) \propto \prod_{t \in Q} P(t|D) \tag{4}$$

Relative document term frequency is a reasonable estimate for $P(t|D)$. However, in this form the model only gives non-zero scores to documents that contain all of the query terms $t_1, ..., t_n$. This corresponds to a strict Boolean model with AND semantics. If a document contains a synonym of a query word instead of one of the exact words then $P(D|Q)$ would be zero because one of the terms is missing and a $P(t|D)$ term is zero. To enable more permissive matching smoothing can be applied where document term frequencies are mediated by a generic model of language from a corpus, C. When all terms are present then the highest scores will result; this is roughly analogous to extended Boolean or coordinate-level ranking. Using linear interpolation, a parameter λ can be introduced to reflect the importance of the term's prescence in the document, Equation 4 can be written:

$$P(D|Q) \propto \prod_{t \in Q} \lambda P(t|D) + (1 - \lambda)P(t|C) \tag{5}$$

It remains to estimate $p(t|C)$. In HAIRCUT the mean relative document term frequency is used for each term t. Regarding λ, it turns out that performance is fairly insensitive

Table 1: Collection Characteristics

	#docs	#unique words	#unique 5-grams	text size (gzip MB)
Bengali	123,040	34,985	1,321,876	151
English	125,516	247,592	839,103	122
Hindi	95,213	19,403	741,915	110
Marathi	99,359	47,940	1,580,775	104

to the precise value selected as long as values near 0 or 1 are avoided [7]. Therefore, while performance could be improved slightly by optimizing choice of λ as a function of tokenization, a smoothing constant of 0.5 was used in all of these experiments. It is also possible to vary λ on a term-by-term basis, but that is not done here.

The training topics (1-25) for the collections were not used in any way.

2. TEXT PROCESSING

A small amount of pre-processing was done with the source data to increase compatibility of the formatting with previous evaluations (e.g., TREC and CLEF) that our software had been tested on. For example, a few document IDs were duplicated and SGML tags were upcased (e.g., "DocNo" was rewritten as "DOCNO").

2.1 Tokenization

The document collections were indexed using several different textual representations, including:

- **words**: space-delimited tokens.
- **4-grams**: overlapping, word-spanning character 4-grams produced from the stream of words encountered in the document or query.
- **5-grams**: length $n = 5$ n-grams created in the same fashion as the character 4-grams.
- **sk41**: overlapping, word-spanning 4-grams and 4-grams that skipped (i.e., omitted) a single interior letter. (See below.)

Common to each tokenization method was conversion to lower case letters, removal of punctuation, and truncation of long numbers to 6 digits.

Lexicon size for several of the collections appeared uncharacteristically small compared to what is typically observed in alphabetic languages. In particular, the number of unique words (i.e., vocabulary size) in Bengali and Hindi seemed improbably low. Unless there was an error in processing, this almost implies that these languages are isolating, or practically devoid of morphological variation. Or that the documents were topically related and lacking in diversity.

2.2 Skipgrams

Consider the present tense conjugation of the Spanish verb contar (*to count*): cuento, cuentas, cuenta, contamos, contáis, and cuentan. Such inflectional variation can cause lexical mismatches that would impair retrieval, and character n-grams are unlikely to be a total solution to this problem since the 1st and 2nd person plural forms do not share longer n-grams with the other forms. However, a regular expression such as *c⋆nt* would match all the related verb forms.

Table 2: Monolingual Runs

	words	4-grams	5-grams	sk41
Bengali	0.1231	0.3280	**0.3582**	0.3352
English	**0.5495**	0.5241	0.5415	0.5264
Hindi	0.0672	0.2820	**0.3487**	0.2746
Marathi	0.1735	**0.3740**	0.3675	0.3478
Average	0.2283	0.3834	0.4040	0.3710

Pirkola et al. [5] have proposed n-grams with skips, using the name *s-grams.*, for matching terminology in cross-language information retrieval between languages sharing a common alphabet. For example, the English word *calcitonin* can be matched to its Finnish translation *kalsitoniini*, supported in part by matches like l⋆t and n⋆n. Mustafa [4] proposed a similar method for monolingual Arabic language processing, where infix morphological changes are common. He identified relevant dictionary terms using bigrams with and without a single skip character and a Dice coefficient to compare sets of bigrams. Järvelin et al. [2] formalized the notion of skipgrams and investigated methods of comparing lexical terms; however, they focused on the case where a single skip is formed by deleting contiguous letters. This makes sense when only bigrams are considered – then the only place to skip characters is between the first and last letters of the (skip) bigram.

But with longer n-grams there are multiple places where skips can occur, and character skipgram methods can be generalized even further by including the possibility of multiple non-adjacent skips within a single word (though no such experiments are reported here). In these experiments skipgrams are considered as an alternative method for tokenization that might support matches across morphologically related words. When a letter is skipped we replace that letter in the n-gram subsequence with a special symbol – a dot character (•). This is done in an attempt to avoid unintended conflations with n-gram strings produced by unrelated words. For the experiments reported here using *sk41* tokenization, the word *cream* would be represented using both regular n-grams *crea* and *ream* in addition to *c•eam*, *cr•am*, and *cre•m*.

3. MONOLINGUAL RESULTS

Our official submissions were based on 4-grams, 5-grams, and sk41 skipgrams using automated relevance feedback. The 10 top-ranked documents were used to generate expansion terms for each query. A different number of expansion terms was used depending on each tokenization method. 50 terms were used with words, 150 terms were used with 4-grams and 5-grams, and 400 terms were used with the skipgrams.

Table 2 lists mean average precision for these runs. The runs using plain words as indexing terms were not officially submitted runs.

Substantial changes were observed based on the language and choice of tokenization used. On average 5-grams were the most effective technique and with 5-grams the best results were obtained in Bengali and Hindi. Words were slightly more effective in English; however, all four indexing schemes performed about the same. 4-grams were slightly better in Marathi compared to 5-grams.

Comparing the anonymized runs released by the organiz-

Table 3: Highest MAP Values

Bengali	English	Hindi	Marathi
0.4719	0.5572	0.3487	0.4483

ers lets us calculate the highest monolingual mean average precision scores for each test set (see Table 3).

The HAIRCUT submissions appear to have the highest performance for Hindi, and competitive performance for English. However, the Bengali and Marathi MAP scores are only about 75% of the top reported runs from those languages.

4. SUMMARY

Several methods of indexing text were compared using the four languages comprising the FIRE test sets. Simple character n-gram tokenization using $n = 5$ appears to give good results when using a statistical language model for retrieval. Skipgram indexing seemed to perform well, but did not outperform regular n-grams. In future work we would like to compare other methods, including statistical approaches to stemming and segmentation.

Also, an analysis of the corpora and resulting indexes should be conducted to ensure no mistakes were made in processing the document collection. The small vocabulary was an unexpected finding.

REFERENCES

[1] Djoerd Hiemstra Using Language Models for Information Retrieval Ph.D. Thesis, University of Twente, 2001.

[2] Anni Järvelin and Antti Järvelin and Kalervo Järvelin. S-grams: Defining Generalized N-grams for Information Retrieval. *Information Processing and Management*, 43(4):1005-1019, 2007.

[3] J. Mayfield and P. McNamee. The HAIRCUT Information Retrieval System. In *Johns Hopkins APL Technical Digest*, 26:1:2-14, 2005.

[4] Suleiman H. Mustafa. Character contiguity in n-gram based word matching: the case for Arabic text searching. *Information Processing and Management*, 41:819-827, 2004.

[5] Ari Pirkola and Heikki Keskustalo and Erkka Leppänen and Antti-Pekka Känsälä and Kalervo Järvelin. Targeted s-gram matching: a novel n-gram matching technique for cross- and mono-lingual word form variants. In *Information Research*, 2(7), 2002.

[6] Jay M. Ponte and W. Bruce Croft A Language Modeling Approach to Information Retrieval In Proceedings of ACM SIGIR 1998, pp. 275-281, 1998.

[7] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst., 22(2):179-214, 2004.