Single N-gram Stemming

James Mayfield and Paul McNamee
The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Rd. Laurel MD 20723-6099 USA
+1 (240) 228-6944 +1 (240) 228-3816
mayfield@jhuapl.edu mcnamee@jhuapl.edu

ABSTRACT

Stemming can improve retrieval accuracy, but stemmers are language-specific. Character n-gram tokenization achieves many of the benefits of stemming in a language independent way, but its use incurs a performance penalty. We demonstrate that selection of a single n-gram as a pseudo-stem for a word can be an effective and efficient language-neutral approach for some languages.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Information Storage and Retrieval – *content analysis and indexing.*

General Terms: Algorithms

Keywords: Stemming, n-gram tokenization.

1. INTRODUCTION

Stemming, an approximation to lemmatization in which morphological variants of a word are reduced to a single form, has been used in information retrieval for decades. In a Boolean system, stemming can be viewed as a recall enhancing device, but in a ranked retrieval system the effect of stemming is not so clear. Nonetheless, stemming remains an active tool in the IR toolbox.

While there are statistical approaches to stemming (*e.g.,* Xu and Croft [9], Jacquemin [4]), the most commonly cited approaches are rule-based (*e.g.,* Porter [7]). Such approaches have a distinct disadvantage in a multilingual setting—new rules must be crafted for each new language to be indexed. While stemming frameworks exist (*e.g.,* SNOWBALL [8]), detailed knowledge of a language is required to use such frameworks to generate a stemmer.

An attractive alternative to stemming is character n-gram tokenization. By indexing using overlapping sequences of n characters, many of the benefits of stemming can be achieved without any knowledge of the target language. This is because some of the n-grams derived from a word will span only portions of the word that do not exhibit morphological variation. For example, the words *juggle, juggling* and *jugglers* share the common 5-gram *juggl.* N-grams work well as indexing terms over a wide variety of languages (see Figure 1). While morphological complexity (as found for example in Arabic, which exhibits infix morphology) may reduce the efficacy of n-gram tokenization, they still perform admirably relative to the use of raw words as indexing terms. N-grams are also entirely language-neutral; no knowledge of a language is required to apply n-gram tokenization to the language beyond selection of a suitable value for n.The drawback of n-grams is not in retrieval accuracy, but rather in retrieval performance and disk usage. Because each character of a text begins a new n-gram, an n-gram representation of a text contains many more indexing terms than does a word or stem representation. Not only does this produce larger indexes, it also increases the number of disk seeks required to locate all of the postings for a query. Techniques for reducing the size of an n-gram index have been proposed [6], but these do not necessarily reduce query execution time. Stemming, although language-specific, does not suffer these performance penalties (the

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 1-58113-646-3/03/0007

Table 1. Document frequencies (CLEF 2002 collection) of 4-grams components of the English word 'juggling'

N-gram	*Document Frequency*	*N-gram*	*Document Frequency*
_jug	681	glin	4,567
jugg	495	ling	55,210
uggl	6,775	ing_	106,463
ggli	3,003		

Table 2. Stems and Pseudo-stems for various words

Language	*Word*	*Stem*	*Pseudo-4*	*Pseudo-5*
English	juggle	juggl	jugg	juggl
English	juggles	juggl	jugg	juggl
English	juggler	juggler	jugg	ggler
English	currency	currenc	rren	rency
English	warren	warren	warr	warre
English	warrens	warren	rens	rrens
Swedish	kontroll	kontroll	ntro	roll_
Swedish	kontrollerar	kontroller	ntro	lerar
Swedish	kontrollerade	kontroller	ntro	lerad
Swedish	kontrolleras	kontroller	ntro	leras

Table 3. Differences significant at the 0.05 level using the Wilcoxon test; differences significant at the 0.001 level are shown in bold.

Comparison	*Significant difference*
4-grams vs. raw words	**de**, **fi**, it, nl, **sv**
4-grams vs. stems	de, fi, sv
4-grams vs. pseudo-stems	**de**, fi, it, nl, **sv**
Stems vs. raw words	de, es, **fi**, fr, it, nl, sv
Stems vs. pseudo-stems	nl
Pseudo-stems vs. raw words	en, **fi**

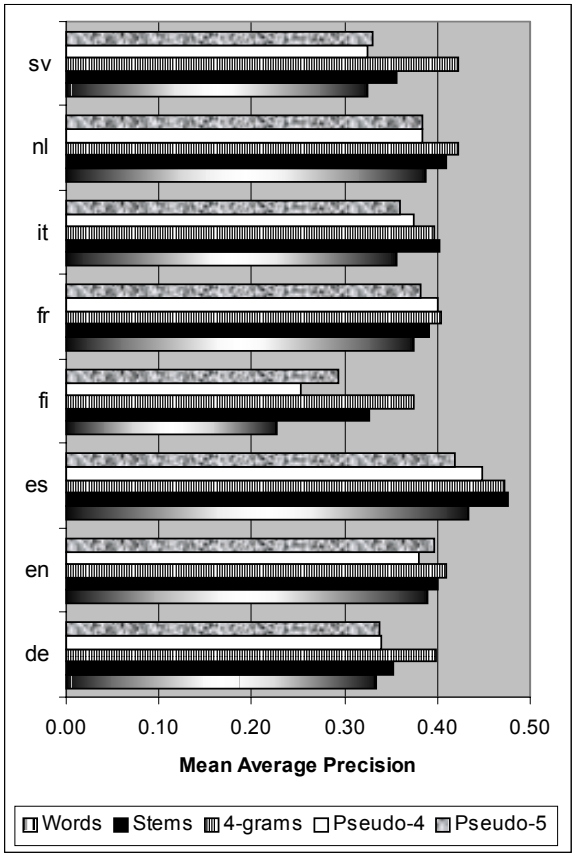

Figure 1. Mean average precision on CLEF 2002 collection for raw words, stems, 4-grams and pseudo-stems.

APPROACH AND EXPERIMENTS

We want to use the simplistic approach to tokenization found with n-grams to simulate stemming in a language-neutral way without paying a concomitant performance penalty. We hypothesize that a single carefully chosen n-gram from each word might serve as an adequate stem substitute. We would like to select an n-gram from the morphologically invariant portion of the word (if such exists). To that end, we select the word-internal n-gram with the highest inverse document frequency (IDF) as a word's pseudo-stem. We reason that affixes indicating a particular morphological variation will be repeated across many different words, and will hence exhibit low IDF. Selecting a single n-gram per term results in an index with the same number of postings as a word or stem index, and in queries that have the same number of terms as for words or stems; thus, the performance penalty paid by n-grams is ameliorated. The technique requires *a priori* knowledge of n-gram frequencies, but calculating such frequencies is straightforward given a monolingual collection in the target language. Table 1 shows document frequencies for the 4-grams that compose the word 'juggling.' Table 2 shows a few examples of words, their Snowball stems and pseudo-stems. Any approach to stemming will exhibit both over-conflation and under-conflation; examples will be seen in the table.

We evaluated the pseudo-stemming technique in eight European languages using the HAIRCUT system [5] and the CLEF 2002 collection [2]. For each language, we measured retrieval accuracy using words, stems, all n-grams, and highest IDF n-gram. Stemmers were from the SNOWBALL project [8]. Our similarity metric used a language modeling approach with Jelinek-Mercer smoothing [3]; no blind relevance feedback was used.

Results expressed in terms of mean average precision are shown in Figure 1. The trend is for 4-grams to perform best, stems next, pseudo-stems next, and raw words lowest; however, individual languages deviate from this ordering. Statistical significance testing (Wilcoxon test at the 0.05 level) is shown in Table 3.

2. CONCLUSIONS

Table 3 shows that some sort of related word conflation has a positive effect on retrieval accuracy for many languages. While n-gram tokenization is almost always preferable to stemming for retrieval accuracy, its associated performance penalty may obviate its use in some settings. Least common n-gram stemming is an attractive solution for a new alphabetic language, because it is so easy to implement and does not incur a performance penalty. Our experiments show improvement in seven out of eight languages over the use of raw words. While the technique generally underperformed stems, this difference was only statistically significant in Dutch; it outperformed stems in French. Further work may lead to better n-gram selection criteria. For example, it may well be that in agglutinative languages the least common n-gram for a compound word will typically span the underlying components. If so, an algorithm that considered other factors (*e.g.,* residual IDF [1]) might lead to higher accuracy. Automatic selection of n-gram length and the use of two or more n-grams for some words are other potentially fruitful directions.

3. REFERENCES

[1] Church, K. W., 'One term or two?' In Fox, E. A., *et al.*, eds., *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-95),* pp. 310-318. 1995.

[2] Cross Language Evaluation Forum Web site. <http://www.clef-campaign.org/>, visited 28 Feb. 2003.

[3] Hiemstra, D., *Using Language Models for Information Retrieval.* PhD Thesis, Center for Telematics and Information Technology, Netherlands. 2000.

[4] Jacquemin, C. 'Guessing morphology from terms and corpora.' In Belkin, N. J., eds., *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-97),* pp. 156-165. 1997.

[5] McNamee, P. Mayfield, J. and Piatko, C. 'The HAIRCUT system at TREC-9.' *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*. NIST Special Publication 500-249, pp. 273-279. 2001.

[6] Pearce, C. E. and Rye, W. E. *N-gram Term Weighting: A Comparative Analysis.* National Security Agency Technical Report #TR-R52-001-98, January 1998.

[7] Porter, M. F., 'An algorithm for suffix stripping.' *Program* 14:130-137. 1980. Reprinted in Sparck Jones, K. and Willett, P., eds., *Readings in Information Retrieval*, Morgan Kaufmann Publishers, pp. 313-316. 1997.

[8] Snowball Web site. <http://snowball.tartarus.org/>, visited 28 Feb. 2003.

[9] Xu, J. and Croft, W. B., 'Corpus-based stemming using co-occurrence of word variants.' *ACM Transactions on Information Systems* 16(1):61-81. 1998.