Statistical vs. Rule-Based Stemming for Monolingual French Retrieval

Prasenjit Majumder[1], Mandar Mitra[1], and Kalyankumar Datta[2]

[1] CVPR Unit, Indian Statistical Institute, Kolkata
[2] Dept. of EE, Jadavpur University, Kolkata
{prasenjit_t,mandar}@isical.ac.in,
kalyandatta@debesh.wb.nic.in

Abstract. This paper describes our approach to the 2006 Adhoc Monolingual Information Retrieval run for French. The goal of our experiment was to compare the performance of a proposed statistical stemmer with that of a rule-based stemmer, specifically the French version of Porter's stemmer. The statistical stemming approach is based on lexicon clustering, using a novel string distance measure. We submitted three official runs, besides a baseline run that uses no stemming. The results show that stemming significantly improves retrieval performance (as expected) by about 9-10%, and the performance of the statistical stemmer is comparable with that of the rule-based stemmer.

1 Introduction

We have recently been experimenting with languages that have not been studied much from the IR perspective. These languages are typically resource-poor, in the sense that few language resources or tools are available for them. As a specific example, no comprehensive stemming algorithms are available for these languages. The stemmers that are available for more widely studied languages (e.g. English) usually make use of an extensive set of linguistic rules. Rule based stemmers for most resource-poor languages are either unavailable or lack comprehensive coverage. In earlier work, therefore, we have looked at the problem of stemming for such resource-poor languages, and proposed a stemming approach that is based on purely unsupervised clustering techniques.

Since the proposed approach does not assume any language-specific information, we expect the approach to work for multiple languages. The motivation behind our experiments at CLEF 2006 was to test this hypothesis. Thus, we focused on mono-lingual retrieval for French (a language which we know nothing about), and tried our statistical stemming approach on French data.

We give a brief overview of the proposed statistical stemming algorithm in the next section. We outline our experimental setup in Section 3, and discuss the results of the runs that we submitted

2 Statistical Stemmer

2.1 String Distance Measures

Distance functions map a pair of strings s and t to a real number r, where a smaller value of r indicates greater similarity between s and t. In the context of stemming, an appropriate distance measure would be one that assigns a low distance value to a pair of strings when they are morphologically similar, and assigns a high distance value to morphologically unrelated words. The languages that we have been experimenting with are primarily suffixing in nature, i.e. words are usually inflected by the addition of suffixes, and possible modifications to the tail-end of the word. Thus, for these languages, two strings are likely to be morphologically related if they share a long matching prefix. Based on this intuition, we define a string distance measure D which rewards long matching prefixes, and penalizes an early mismatch.

Given two strings $X = x_0x_1 \ldots x_n$ and $Y = y_0y_1 \ldots y_{n'}$, we first define a Boolean function p_i (for penalty) as follows:

$$p_i = \begin{cases} 0 & \text{if } x_i = y_i \quad 0 \leq i \leq \min(n, n') \\ 1 & \text{otherwise} \end{cases}$$

Thus, p_i is 1 if there is a mismatch in the i-th position of X and Y. If X and Y are of unequal length, we pad the shorter string with null characters to make the string lengths equal.

Let the length of the strings be $n + 1$, and let m denote the position of the first mismatch between X and Y (i.e. $x_0 = y_0,\ x_1 = y_1,\ \ldots,\ x_{m-1} = y_{m-1}$, but $x_m \neq y_m$). We now define D as follows:

$$D(X,Y) = \frac{n-m+1}{m} \times \sum_{i=m}^{n} \frac{1}{2^{i-m}} \quad \text{if } m > 0, \quad \infty \text{ otherwise} \tag{1}$$

Note that D does not consider any match once the first mismatch occurs. The actual distance is obtained by multiplying the total penalty by a factor which is intended to reward a long matching prefix, and penalize significant mismatches. For example, for the pair $\langle astronomer, astronomically \rangle$, $m = 8, n = 13$. Thus, $D_3 = \frac{6}{8} \times (\frac{1}{2^0} + \ldots + \frac{1}{2^{13-8}}) = 1.4766$.

2.2 Lexicon Clustering

Using the distance function defined above, we can cluster all the words in a document collection into groups. Each group, consisting of "similar" strings, is expected to represent an equivalence class consisting of morphological variants of a single root word. The words within a cluster can be stemmed to the 'central' word in that cluster. Since the number of natural clusters are unknown apriori, partitive clustering algorithms like k-means are not suitable for our task. Also, the clusters are likely to be of non-convex nature. Graph-theoretic clustering

algorithms appear to be the natural choice in this situation because of their ability to detect natural and non-convex clusters in the data.

Three variants of graph theoretic clustering are popular in literature, namely, *single-linkage*, *average-linkage*, and *complete-linkage* [2]. Each of these algorithms are of hierarchical (agglomerative or divisive) nature. In the agglomerative form, the cluster tree (often referred to as a dendogram) consists of individual data points as leaves. The nearest (or most similar) pair(s) of points are merged to form groups, which in turn are successively merged to form progressively larger groups of points. Clustering stops when the similarity between the pair of closest groups falls below a pre-determined threshold. Alternatively, a threshold can be set on the distance value; when the distance between the pair of nearest points exceeds the threshold, clustering stops. The three algorithms mentioned above differ in the way similarity between the groups is defined. We choose the compete-linkage algorithm for our experiments.

3 Results

We used the Smart [3] system for all our experiments. We submitted four official runs, including one baseline. For the baseline run (Cbaseline), queries and documents were indexed after eliminating stopwords (using the stopword list provided on the CLEF website[1]). The `<title>`, `<desc>`, and `<narr>` field of the query were indexed. The *Lnu.ltn* term-weighting strategy [1] was used. No stemming was done for the baseline run.

For the remaining three runs, we used three variants of the statistical stemming method described above. Since our approach is based on hierarchical agglomerative clustering (as described above), the threshold value used in the clustering step is an important parameter of the method. Earlier experiments with English data have shown that 1.5 is a reasonable threshold value. We generated two retrieval runs by setting the threshold to 1.5 and 2.0 respectively (Cd61.5, Cd62.0).

For the third run, the stemmer was created based on a subset of the data. A lexicon was constructed using only the LeMonde section of the document collection, and this was then clustered as described above to determine the stem classes. Since the lexicon was smaller, the clustering step took less time for this run. The motivation behind this experiment was to study how performance is affected when a subset of the lexicon is used to construct the stemmer in order to save computation time.

After the relevance judgments for this data set were distributed, we performed two additional experiments: first, we tried setting the clustering threshold to 1.0; and secondly, we used the French version of Porter's stemmer[2] in place of our statistical stemmer. The results obtained for all the official and unofficial runs are given below.

[1] http://www.unine.ch/info/clef/

[2] Downloaded from `http://www.snowball.tartarus.org/algorithms/french/stemmer.html`

Table 1. Retrieval results obtained using various stemmers

Run ID	Topic fields	MAP	Rel. ret.	R-Precision
Cbaseline	T+D+N	0.3196	1,616	31.30%
Cd61.0	T+D+N	0.3465 (+8.4%)	1,715	34.85%
Cd61.5 (official)	T+D+N	0.3454 (+8.1%)	1,709	34.35%
Cd61.5 (obtained)	T+D+N	0.3509 (+9.8%)	1,708	34.26%
Cd62.0	T+D+N	0.3440 (+7.6%)	1,737	34.78%
Cld61.5	T+D+N	0.3342 (+4.6%)	1,678	32.81%
Porter	T+D+N	0.3480 (+8.9%)	1,705	34.71%

The official results for the run labelled Cd61.5 do not agree with the evaluation figures that we obtained by using the distributed relevance judgment data. We therefore report both the official figures and the numbers that we obtained in Table 1.

The two most promising runs (Cd61.5 and Porter) were analyzed in greater detail. Paired t-tests show that stemming (using either strategy) results in significant improvements (at a 1% level of confidence) over the baseline (no stemming), but the differences between the rule-based and statistical approaches are not statistically significant. Also, some loss in performance results when the stemmer is generated from a subset of the corpus (run Cld61.5).

This confirms our hypothesis that the proposed stemming approach, which does not assume any language-specific information, will work for a variety of languages, provided the languages are primarily suffixing in nature.

References

1. Buckley, C., Singhal, A., Mitra, M.: Using Query Zoning and Correlation within SMART: TREC5. In: Voorhees, E.M., Harman, D.K. (eds.) Proceedings of the Fifth Text REtrieval Conference (TREC-5). NIST Special Publication, pp. 500–238 (November 1997)
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. 31(3), 264–323 (1999)
3. Salton, G.: The SMART Retrieval System—Experiments in Automatic Document Retrieval. Prentice Hall Inc., Englewood Cliffs (1971)