Hungarian and Czech Stemming using YASS

Prasenjit Majumder
Mandar Mitra
Dipasree Pal
CVPR Unit, Indian Statistical Institute, Kolkata
{prasenjit_t,mandar,dipasree_t}@isical.ac.in

Abstract

This is the second year in a row we are participating in CLEF. Our aim is to test the performance of a statistical stemmer on various languages. Last year, we tried the stemmer on French; this year, we opted for Hungarian, Bulgarian and Czech. We were unable to complete the Bulgarian task, but submitted official runs for the ad-hoc monolingual Hungarian and Czech tasks. We find that, for both languages, the performance of the statistical stemmer is comparable to that of an available rule-based stemmer.

1 Introduction

Stemming is arguably a recall enhancing device in text retrieval. Most commonly used stemmers are rule-based and therefore language specific. Such stemmers are unlikely to be available for resource-poor languages. In earlier work, therefore, we proposed YASS, a statistical stemmer. As YASS 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. Since our hypothesis was supported by last year's experiments, this year, we planned on mono-lingual retrieval for three more languages which we know nothing about.

The main stumbling block in our experiments was the encoding issue. For all our experiments, we use the SMART system, which is not Unicode compatible. Our major task, therefore was to incorporate UTF-8 support in SMART.

We give a brief overview of YASS in the next section. Section 3 describes the training experiments of YASS which is followed by the results of all the runs (both official and unofficial) in for this year in the Section 4.

2 YASS

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*. 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 Experiments with Hungarian and Czech

We have mentioned earlier that YASS needs no linguistic input as it is a statistical stemmer. However, before running YASS on a new language we need to train it. For training we need a corpus of that language. In the following subsections, we will describe the training procedure of YASS in brief.

3.1 Training

A lexicon is extracted from a given language corpus and clustered. The clustering threshold is learned from the training data. For Hungarian we used 2006 CLEF data for training. For Czech we did not get previous years data and thus we did not train our stemmer for Czech.

3.1.1 Hungarian

The same Hungarian corpus is used for the 2005, 2006, and 2007 tasks. The lexicon extracted from the corpus has 536678 surface words. The lexicon was clustered using various threshold settings, and the number of clusters versus threshold curve is shown in Figure 1. The step like regions around 0.8, 1.1, 1.5, 2.0 suggest that the number of clusters is stable around these threshold values. These values may thus be chosen as candidate thresholds for clustering.

After clustering the lexicon using these four threshold values, the lexicon size gets reduced to 225489, 169619, 130278, 76782 classes respectively. The stemmers thus prepared are used in four different runs.

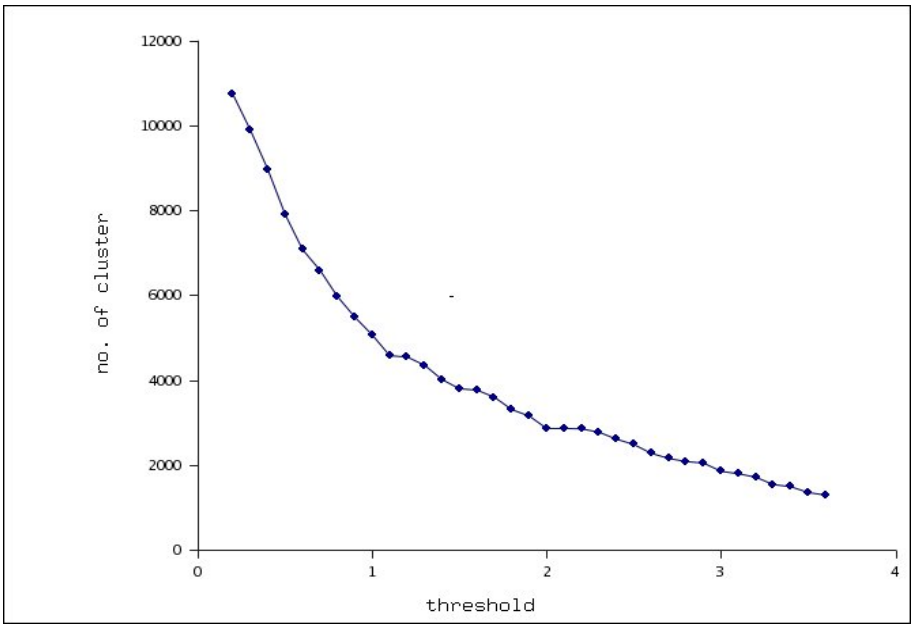

Figure 1: Threshold vs. no of cluster for Hungarian

The official topics for the Hungarian monolingual run at CLEF-2006 were topic numbers 301 to 325 and 351 to 375. We find that YASS performs best at the threshold 1.1. A mean average precision (MAP) versus threshold curve for all the runs are plotted and given in Table 1.

Threshold	MAP
0.8	0.2692
1.1	0.2835
1.5	0.2777
2	0.2735

Table 1: Threshold vs. MAP for Hungarian

We tested these stemmers on CLEF queries 251 to 300. These queries were used in the CLEF 2005 monolingual Hungarian task. The highest MAP obtained here was again at $\theta = 1.1$. We tried to compare our results with that of Tordai et al.[3], but could not compile the stemmer used by them. We have therefore reproduced the same experiments using YASS. Table 2 compares our results with those reported by Tordai et al. Tordai et. al report the results of 8 runs in their paper, out of which we listed the best three runs (*4-Gram*, *Heavy minus hyphen* and *5-Gram*).

YASS			
Run Name	**MAP**	**R-prec**	**% Rel_Ret**
noStem(T+D+N)	0.2472	0.2531	72.8434
θ = 0.8(T+D+N)	0.3211	0.3231	83.8125
θ =1.1(T+D+N)	0.3246	0.3247	86.0489
θ =1.5(T+D+N)	0.3246	0.3190	86.6879
θ =2.0(T+D+N)	0.3246	0.3068	83.8125
noStem(T+D)	0.2170	0.2285	69.1160
θ =0.8(T+D)	0.3121	0.3162	81.0436
θ =1.1(T+D)	0.3241	0.3270	84.2385
θ =1.5(T+D)	0.3268	0.3309	85.6230
θ =2.0(T+D)	0.3048	0.3074	84.1320
TORDAI et. al			
Run Name	**MAP**	**R-prec**	**% Relevant Docs Retrieved**
Heavy minus hyphen	0.3099	0.3048	83.1
4-Gram	0.3303	0.338	83.6
5-Gram	0.3002	0.3057	82.4

Table 2: Results of previous Hungarian runs

3.1.2 Czech

For Czech no training data was available. The threshold for the Czech stemmer was set to 1.5 based on our experience with English and French.

4 Results of CLEF 2007

Unfortunately, we could not complete the training experiments described above, before the official submission. Thus for both the languages, we have submitted three runs, untrained.

In the first Hungarian run *ISI.YASSTDHUN*, we indexed only the `<title>` and `<desc>` fields of the queries. For the second run, *ISI.YASSHUN* we indexed the `<title>`, `<desc>`, and `<narr>` fields of the queries. In both cases the clustering threshold was set to 1.5. For the third run, *ISI.ISIDWLDHSTEMGZ*, we made use of the Hungarian stemmer available from the web [1].

The Czech runs are analogous: the first run uses only the `<title>` and `<desc>` fields; the second and third runs use the complete query. The second run makes use of an existing stemmer[2] instead of YASS. The final run was a baseline run where no stemming was used.

After the relevance judgments for the data sets were distributed, we performed some additional experiments for both the languages. The results obtained for all the official and unofficial runs are given in Table 3 and 4. These results confirms our hypothesis that YASS will work for a variety of languages, provided the languages are primarily suffixing in nature.

Our ignorance of the languages prevents us from doing a detailed post-mortem on these results. For the benefit of those who understand Hungarian / Czech, we provide some examples of words and their roots (as obtained using YASS) in the following table. These words were selected from queries on which the stemmed run did significantly better than the unstemmed run.

References

[1] C. Buckley, A. Singhal, and M. Mitra. Using Query Zoning and Correlation within SMART: TREC5. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238, November 1997.

[1] http://snowball.tartarus.org/algorithms/hungarian/stemmer.html
[2] http://members.unine.ch/jacques.savoy/clef/index.html

[2] Gerard Salton, editor. *The SMART Retrieval System—Experiments in Automatic Document Retrieval*. Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[3] Anna Tordai and Maarten de Rijke. Four stemmers and a funeral: Stemming in hungarian at clef 2005. In *CLEF*, pages 179–186, 2005.

Hungarian Runs Submitted (*nnn.ltn*)			
Run Name	**MAP**	**R-prec**	**% Rel_Ret**
ISI.YASSHUN	0.1712	0.1974	72.22
ISI.YASSTDHUN	0.1695	0.1943	72.88
ISI.ISIDWLDHSTEMGZ	0.1605	0.1858	66.84
Other Hungarian Runs (*Lnu.ltn*)			
Run Name	**MAP**	**R-prec**	**% Rel_Ret**
noStem(T+D+N)	0.2647	0.2849	71.59
hun.d6-0.8(T+D+N)	0.3276	0.3316	81.11
hun.d6-1.1(T+D+N)	0.3459	0.3437	84.19
hun.d6-1.5(T+D+N)	0.3600	0.3565	84.96
hun.d6-2.0(T+D+N)	0.3638	0.3747	83.42
noStem(T+D)	0.2320	0.2571	65.53
hun.d6-0.8(T+D)	0.2982	0.3084	76.83
hun.d6-1.1(T+D)	0.3094	0.3095	79.91
hun.d6-1.5(T+D)	0.3203	0.3274	81.22
hun.d6-2.0(T+D)	0.3638	0.3747	75.08

Table 3: All Hungarian runs on 2007 CLEF data

Runs Submitted (*Lnu.ltn*)			
Run Name	**MAP**	**R-prec**	**% Rel_Ret**
ISI.CZTD [YASS] (T+D)	0.3224	0.3102	87.13
ISI.ISICL [dnlded] (T+D+N)	0.3362	0.3326	89.37
ISI.ISICZNS [nostem] (T+D+N)	0.2473	0.2540	76.64
Other runs (*Lnu.ltn*)			
Run Name	**MAP**	**R-prec**	**% Rel_Ret**
CzeTDN.cze.d6-1.1.Lnu (T+D+N)	0.3305	0.3208	89.63
CzeTDN.cze.d6-1.5.Lnu (T+D+N)	0.3390	0.3264	89.23
CzeTDN.cze.d6-2.0.Lnu (T+D+N)(Lnu)	0.3381	0.3213	89.89

Table 4: All Czech runs on 2007 CLEF data

Hungarian		Czech	
politikusokról, politikai	politi	Kosteličových, Kosteličovi	Kostelič
atomhulladékot	atomhulladék	prezidenští, prezidenta, prezidentského	preziden
megszűnése	megsz	kandidáti, kandidáta	kandidát
elnökjelöltek, elnökjelölt	elnökjelöl	vesmírní, vesmírných, vesmíru	vesmír
királynő, királyságbeli	kir / király	turistech, turisté	turist

Table 5: Word stems generated by YASS