UNIVERSITY OF ZAGREB, FACULTY OF ELECTRONICS AND COMPUTING

Computing optimal tresholds for q-gram filters

Stipe Kuman, Dino Radaković, Leon Rotim

January 13, 2016

1 INTRODUCTION INTO THE PROBLEM

In this project we have analyzed and recreated a paper on the topic of finding optimal thresholds for q-gram filters. Q-gram filters are used for matching substrings of length q in a text with substrings in a given pattern. In addition, the paper we are recreating uses gapped q-grams, which allow for discontinuous text substrings. Gapped q-grams were shown to be more efficient than contiguous q-grams in the right conditions. However, computing an optimal threshold for those filters is also more difficult. Our project was done in C++ and can run on the bio-linux platform. In the next section we will describe the process we used in our computation. Before that, we will describe some of the main principles our algorithm is based on.

1.1 PROBLEM DESCRIPTION

First off, we will describe the notation used in this assignment. T is the text file, P is the given pattern and S stands for all the substrings in T that match the given pattern P within a Hamming distance of k (number of non-matching characters). The length of the q-gram filter is q and the threshold is represented by t. Finally, m is the length of P and S.

The goal of the q-gram filter is to reduce the number of potential matches that need to be compared to the given pattern P. The threshold variable t defines the minimum number of q-gram matches between the pattern P and a substring so that the substring would be considered as a potential match. A low value of t will result in too many potential matches, while a too high value may overlook some potential matches which would make the filter lossy. Only filters that label all the matches as potential matches will be considered in the assignment.

As stated previously, this assignment will focus on gapped q-grams. For contiguous q-grams an optimal threshold can be computed with a simple mathematical formula, which is not true for gapped q-gram filters. In gapped q-gram filters the problem is much more complex because a closed form formula has not been found. Because of that we have used the dynamic algorithm found in the assignment paper.

1.2 Pruning method

Even with this reasnobly fast dinamic programming algorithm computing all possible gapped Q-grams with positive threshold for parameters $m = 50$, $k = \{4,5\}$ is still a challenging task. Using brute force method, calculating threchold for all shapes in that have $m = 50$ and $k = \{4,5\}$ and leaving only those for which positive threshold is computed, would leave us with search space of 2^{50} thresholds to compute. Since that kind of search space is clearly unfeasable we use fallowing lemma as stated in [**?**].
If $Q' \subseteq Q$ then $t_{Q'}(m,k) \geq \mathrm{t}_Q(m,k)$
Using lemma 1.2 we propose simple pruning technique which we used to compute all positive thresholds. Say that we have a set $Set_{current}$ which contains all Q-grams with size q, and we know values of thresholds for each Q-gram in the set. Next we define two empty sets, Set_{next} and $Set_{forbbiden}$, and for each Q-gram, Q_i in $Set_{current}$ run the following recurrence.