Text Technologies for Data Science Assignment 1

s1140740

1 Introduction

The purpose of this report is to outline decisions made in the implementations of search algorithms for information retrieval and discuss the performance on provided datasets and techniques used.

2 Overlap & tf.idf algorithms

Both of these algorithms have a fairly similar structure. Queries and documents are tokenized and turned into a vector space model.

2.1 Overlap algorithm

This algorithm counts the number of words that were present in both the query and the document. The mean average precision (MAP) of this search algorithm was 0.1527.

2.2 tf.idf algorithm

This algorithm includes a pre-processing step where the total number of documents, tokens and average document length were computed. Moreover, we compute a dictionary of word to count mappings for the whole collection. These are later used together with the query and document dictionaries to compute the tf.idf value. The MAP of this search algorithm was 0.3248.

3 My best algorithm

My best algorithm combines multiple techniques. It uses tf.idf weighing from the previous task, Porter stemmer from snowball, and the Chi-square statistic. I have tried multiple other techniques before arriving at this solution.

Firstly, I tried splitting tokens into character n-grams. This lowered the MAP to 0.3197. The snowball stemmer worked better and improved MAP to 0.3479 when using the original porter algorithm and 0.3482 when using the improved porter2 algorithm.

Secondly, I tried tuning parameter `k` used in frequency normalisation. I achieved highest MAP 0.3492 with `k` equal to 5. The improvement was minimal. Additionally, I tried removing stop words but again that improved MAP only marginally to 0.3567, as expected since the tf.idf algorithm gives little weight to stop words.

Thirdly, I tried maximum tf normalization which normalises the tf weights of all terms occuring in a document by the maximum tf in that document. This normalisation lowered MAP to 0.25 at best (when paramater `a` was set to 0.4, as suggested in chapter 6.4.2 of Introduction to Information Retrieval).

Finally, I ran tests with the following statistical synonym measures: Mutal Information (MIM), Expected Mutual Information (EMIM), Dice's coefficient and Chi-square. The results from testing these measures combined with using and not using a stemmer can be seen in table 1. Chi-square and the porter stemming algorithm provided the best result even though the difference between stemmers was minimal.

Measure	Porter	English	No Stemmer
Chi-square	**0.4525**	0.4511	`0.4470`
Dice	0.3768	0.3646	`0.4233`
MIM	0.3285	0.3252	`0.3088`
EMIM	0.1469	0.1436	`0.1604`
No Measure	0.3479	0.3482	`0.3248`

Table 1: MAP for different stemmer algorithms and statistical measures