Text Technologies for Data Science: Assessment 1

s1107496

October 6, 2014

1 Introduction

This report describes the implementation of two information retrieval algorithms, overlap.py and tfidf.py, as well as refinements to the latter as implemented in best.py.

2 overlap.py

In overlap.py, each query and document was first transformed to lowercase and tokenized, with tokens matching the simple regular expression `[A-Za-z0-9]+`. For every combination of document and query, the score returned was the size of the intersection of the token sets for the document and query. This method resulted in an average precision of 0.1527.

3 tfidf.py

tfidf.py implemented the standard tf-idf algorithm, with k=2 and using the natural logarithm. For speed, df-scores for each token were pre-calculated. The tf-idf algorithm resulted in an average precision of 0.3248.

4 best.py

best.py included a number of refinements over tfidf.py, as listed below. The average precision achieved by best.py was 0.3508.

4.1 Stop word removal

After initial tokenization using the same regular expression as overlap.py, all tokens from the English-language stop word list provided by `nltk` were removed. This was performed for both queries and documents, and increased average precision by 0.0088.

4.2 Character n-grams

To account for morphological variation, each token was broken into constituent n-grams, which were then themselves added to the token list. This procedure was performed for both queries and documents. As stated in the 29th September, 2014 Text Technologies lecture, character n-grams of length 4 or 5 work best for European languages - in this case, 4-grams yielded an average precision advantage of 0.003 over 5-grams. Adding both 4- and 5-grams decreased average precision. Overall, this method increased average precision by 0.0066.

4.3 Bigrams

In both documents and queries, to capture meaningful pairings of words, bigrams from the stop-word-sanitized token list were themselves added to the token list. This improved average precision by 0.0033, and was found to be more effective than trigrams or both bigrams and trigrams.