11791 HW3 Report

Anurag Kumar
Andrew id: alnu@andrew.cmu.edu

October 21, 2014

1 Task 2

1.1 Error Analysis

On analysing each query I found following error types. Several of the error types occurred at the same time. Types of errors found were

- Stemming Problem - For some queries, the similarity between query and the relevant document reduced because of stemming problem. Words like *die* and *died* were treated as different entities. I found this error for **3** queries.
- Extra/Irregular Information - In several cases the most relevant document contained unnecessary extra information. The extra informations resulted in reduction of similarity measure. This error also includes cases where similar but not exact words were used. Example is *spaceship* and *spacecraft*. This error occurred for **10** queries.
- Tokenization Problem - The basic system tokenizes only on white spaces. This also causes error. This error occurred for **2** queries.
- Case: Casing causes problems in atleast **2** queries
- Polysemy - Same word referring to different objects. This error occurred in **1** query.

1.2 Design Analysis

The first part of design is Type System. I used the provided type system without any changes. It consists of two types a "Document" type and "Token" type. The "Document" type has attributes to store qId, relevance, token lists and document text. The "Token" type system stores the text and its frequency. An aggregate analysis engine which runs all three engines of the pipeline has been used. The pipeline includes 3 important engines, namely Document Reader, Document Vector Annotator and Retrieval Evaluator. Document Reader reads each line of the document and sets the values to attributes in "Document" type system. This has already been provided and I did not made any significant change to the provided code. DocumentVectorAnnotator is used for creating the frequency vector for each term in the document. The first step in this process is tokenization which has been done on white spaces. The method createTermFreqVector in DocumentVectorAnnotator.java then implements the term vector generation. Frequency and text attributes for the "Token" type system and tokenlist in "Document" are updated in JCAS. In the RetrievalEvaluator the tokens and frequency are stored as HashMaps. Three array lists corresponding to qId, rel value and document string are maintained to store these values. This solves the problem of inputs coming from different CASes. For each query documents its similarity measure with other documents are computed and stored in an Array List. A separate Array List is maintained for holding the ranks of relevant document for each query. From the perspective of machine learning, similarity or distance measures is the most important aspect. Several distance measures apart from those mentioned in the assignment can be tried but the dataset is too small to make a concrete conclusion about which is best. Moreover, the baseline does tokenization on white spaces. Text normalization such as conversion to lower case, punctuation removal thus become important.

1.3 Improvements

Method	MRR
Cosine Similarity	0.4375
Cosine+ Case Handled	0.4583
Cosine+ Case Handled +period removed	0.4916
Jacard+ Case Handled +period removed	0.4625
Euclidean Distance+Case Handled +period removed	0.6375