11-693 Software Method for Biotechnology Homework 4
Name: Tao Guo
id: taog

Report of Engineering and Error Analysis with UIMA

1. Requirement analysis

Due to the assignment, the information processing pipeline consists of 5 steps.

1. correctly extract bag of words feature vector from the input text collection
2. compute the cosine similarity between two sentences in the text collection
3. compute the Mean Reciprocal Rank (metric) for all sentences in the text collection 4. Design a better retrieval system that improves the MRR performance measure
5. Improve the efficiency of the program by doing error analysis of retrieval system

2. Logical Data Model

The logical data model design is illustrated as the following UML type diagram (figure).

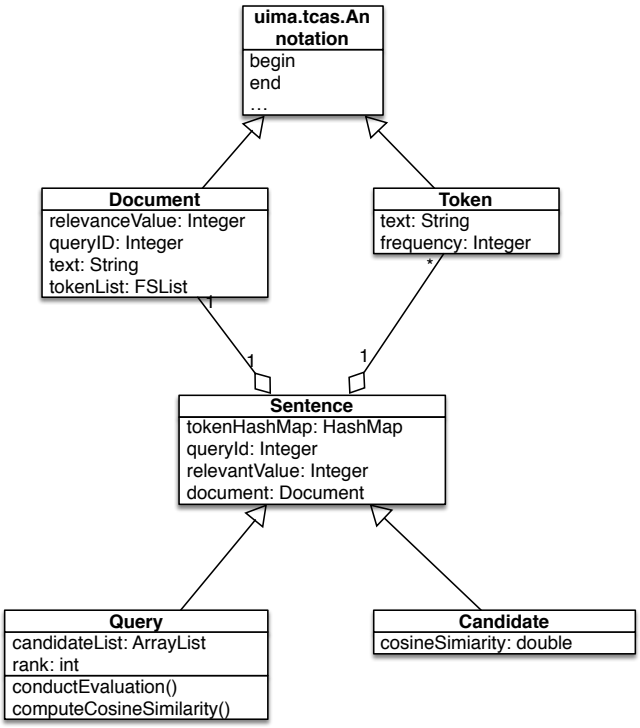

Figure: UML Type Diagram of the Design

The type system contains two different types. *Document* and *Token*.

2.1 Type *Document*
In this project, one *document* object denotes to one sentence. It has the *relevanceValue*, *queryID*, *text* and *tokenList* features to record the corresponding information of the document.

2.2 Type *Token*
The type *Token* denotes to a token in the document. It has the *text* and *frequency* features to record the string of the token as well as the number of times it appears in the document.

3. Analysis Engine Implementation

In the analysis engine implementation, the system contains 3 parts. That is annotators, CAS Consumers and collection readers.

3.1 Collection Readers
The collection reader contains one class *DocumentReader.java,* which is readily implemented in the prototype. It parses the data line, splitting the sentence using tab and extract the value of queryID, relevance value and text.

3.2 Annotators
The annotator system contains only one annotator *DocumentVectorAnnotator.java*. In this annotator, an HashMap object *tokenCountMap* is utilized to count the frequency of tokens efficiently. A lot of Token instances are created through according to this HashMap and referred in the *TokenList* Finally, convert the *TokenList* to an instance of FSList to be stored in the CAS document.

3.3 CASConsumers
The CASConsumers contains one consumer class *RetrievalEvaluator.java* and three additional classes. As all the CAS document instances will be destroyed after the *processCas* method, but some of the evaluation part should be done after that, the consumer class created several objects to store the information and logical relations of the documents. The evaluation will be done in the *collectionProcessComplete* method the HashMap *queryHashMap* is used to store the *queryId* and its corresponding *Query* object.

The *Sentence* class is the abstract class for the *Query* and *Candidate* class. It provides the *tokenHashMap*, *queryID*, *relevantValue* and *document* reference features for each sentence.

The *Query* class represents a query sentence. In addition to the features provided by the *Sentence* abstract class, this class implements the *candidateList* to store reference to the corresponding candidate sentences. And the *rank* field denotes to the reciprocal rank of the correct candidate. The method to compute cosine similarity and the rank *conductEvaluation()* is also implemented in this class.

The *Candidate* class denotes to a candidate sentence of the query. The *cosineSimilarity* field is provided to store the cosine similarity score.

4. Error Analysis

4.1 Straight Forward Approach

At first, I implemented the system directly without considering any optimization and refinement. I splits the tokens using space and just do the computing and evaluation. The data generated in the straight forward approach is shown as follows in Table 1.

qid	rel	Score	Rank	Text
1	99		1	Classical music is dying
1	0	0.267		Pop music has absorbed influences from most other genres of popular music
1	1	0.452		Classical music may never be the most popular music
1	0	0.177		Everybody knows classical music when they hear it
2	99		1	Energy plays an important role in climate change
2	0	0.000		Old wine and friends improve with age
2	0	0.000		With clothes the new are the best, with friends the old are the best
2	1	0.102		Climate change and energy use are two sides of the same coin.
3	99		1	One's best friend is oneself
3	1	0.507		The best mirror is an old friend
3	0	0.434		My best friend is the one who brings out the best in me
3	0	0.183		The best antiques are old friends
4	99		3	The shortest distance between new friends is a smile
4	0	0.236		Wear a smile and have friends; wear a scowl and have wrinkles
4	1	0.172		If you see a friend without a smile, give him one of yours
4	0	0.287		Behind every girls smile is a best friend who put it there
5	99		3	It takes a long time to grow an old friend
5	0	0.000		Old wine and friends improve with age
5	0	0.060		With clothes the new are the best, with friends the old are the best
5	1	0.000		Old friends are best

Table 1: Data generated in straight forward approach

The first three data sets seems to run really perfect. However, in the last two data sets it doesn't perform well. Some words are capitalized because they are the first word of the sentence. For example, the word "Old" in query 5. In that case, they are not matched. At the same time, the punctuation really affects the tokenization, making the scoring more inaccurate.

4.1 Optimized Approach

Due to the problems occurs in the first step. I optimized the algorithm by lower-casing the tokens before matching and deleting the extra spaces and punctuation before and after the token. The data is as follows in Table 2.

qid	rel	Score	Rank	Text
1	99		1	Classical music is dying
1	0	0.267		Pop music has absorbed influences from most other genres of popular music
1	1	0.452		Classical music may never be the most popular music
1	0	0.354		Everybody knows classical music when they hear it
2	99		1	Energy plays an important role in climate change
2	0	0.000		Old wine and friends improve with age
2	0	0.000		With clothes the new are the best, with friends the old are the best
2	1	0.306		Climate change and energy use are two sides of the same coin.
3	99		1	One's best friend is oneself
3	1	0.507		The best mirror is an old friend
3	0	0.434		My best friend is the one who brings out the best in me
3	0	0.183		The best antiques are old friends
4	99		3	The shortest distance between new friends is a smile
4	0	0.298		Wear a smile and have friends; wear a scowl and have wrinkles
4	1	0.258		If you see a friend without a smile, give him one of yours
4	0	0.289		Behind every girls smile is a best friend who put it there
5	99		1	It takes a long time to grow an old friend
5	0	0.119		Old wine and friends improve with age
5	0	0.056		With clothes the new are the best, with friends the old are the best
5	1	0.158		Old friends are best

Table 2: Data generated in straight forward approach

As is shown in Table 2. The performance on the 5th data set is largely improved. And if casing analysis on each score, the confidence of right answer is greatly improved, which also indicate the overall improvement in the performance.

4.3 Further improvement

I tried to delete the stop words in the document before casing matching and computing. But it didn't improve the performance significantly. In order to maintain great efficiency when handling large data set, I tried to utilize HashMap whenever necessary to reduce the time complexity of the algorithm.