11-791: Design and Engineering of Intelligent Information System

Homework 3

Zexi Mao
zexim@andrew.cmu.edu

October 21, 2014

1 Architecture

1.1 Type system

The type system for the gene mention tagging task consists of the following types:

- `Document`, which extends `org.apache.uima.jcas.tcas.Annotation`, is a type for documents (in the case of this homework they are sentences). One feature to notice under this type is `tokenList`, it is an `FSList` of `Token`s. This list is the sparse term vector of a document.
- `Token`, which also extends `org.apache.uima.jcas.tcas.Annotation`, is a type for the terms. Each `Token` is document specific, it stores the term text and term frequency in the corresponding document.

1.2 Annotator

The Annotator in Homework 3 is implemented with the class `DocumentVectorAnnotator`. In this Annotator, each of the `Document` is tokenized and put into the sparse term vector.

1.3 CAS consumer

The CAS consumer in this task is the most important part. It is implemented with the class `RetrievalEvaluator`. In this class, various data structures are used to store information needed to rank the query results and generate reports.

A private class `ReportEntry` is used in this class for storing the information needed for the report. For each query, one `ReportEntry` is generated, the information in it includes the gold standard relevant document index, its score, its rank, and scores for all documents in this query.

2 Error Analysis

In the 20 queries, the retrieval system in Task 1 only got 1 correct (rank the relevant document the first). For the other 19 queries, the following table shows their error category.

Table 1: Error Classification

Tokenization or stemming error	6
Letter case issue	1
Key information issue	7
Phrasing difference	5

The improvement I made for this system is mostly in the stemming algorithms. Firstly, the tokens are splitted not only based on white spaces but also on punctuations. After the tokenization, lematization is performed using the Stanford Lemmatizer (the lemmatizer also does automatic lower case for all English characters). Stopword removal is performed after that. By using the new feature vector representation, the MRR improved from 0.4375 to 0.525.

The code for Task 2 is available at `https://github.com/KyleMao/hw3-zexim/tree/task2`.