Ryan Carlson
11-791, Homework 4
October 28, 2013

This assignment had two tasks:

1. To fill in the stubs of code, implementing cosine similarity, ranking results along that metric, and computing MRR; and
2. To study the mistakes made by cosine similarity and devise a better method to rank the answers.

Task 1: Baseline

To accomplish this first step, I first implemented cosine similarity. Then I realized I wanted a nice way to sort documents by their scores and to keep all the relevance information handy, so I created a class RetrievalEvaluationDocument that contains the following information

- query identifier
- whether or not the document represents the query
- whether or not the document represents the correct answer
- the text of the document (useful for debugging)
- the word frequencies, stored as a String -> Integer Map
- and a score

RetrievalEvaluationDocument implements the Comparable interface, so two documents can be compared by their scores.

Extracting the necessary information from each Document object makes it very easy to sort documents once they've been assigned scores. Tokens were extracted using Stanford NLP's TokenizerFactory. I did not change the type system.

The baseline system using only cosine similarity has the following output

```
score: 0.45226701686664544  rank=1     rel=true qid=1
score: 0.294174202707276    rank=1     rel=true qid=2
score: 0.4629100498862757   rank=2     rel=true qid=3
 (MRR) Mean Reciprocal Rank ::0.8333333333333334
Total time taken: 1.422
```

Note that the MRR was .83, and the system took 1.4 seconds total.

Part 2: Error Analysis & Improvement

So, what was wrong with this system? Well, the first two questions were ranked correctly, so this cosine similarity metric must be on to something! So let's focus on what went wrong with the third question.

- The query was: `One's best friend is oneself`
- The gold standard answer was: `The best mirror is an old friend`
- The answer we chose was: `My best friend is the one who brings out the best in me`

We chose this answer because it shared more words in common with the query than the gold standard answer did. There were two instances of "best" and a single instance of "one" in our top-rated answer.

In other classes we've discussed linearly interpolating the scores of two different systems to improve it, and I decided that would be an interesting next step. The question was, what should this next system look like? What kind of indicator could I use to distinguish the gold standard from the rest, without impacting the other queries?

Looking at the [query, answer] pairs, I started to notice that the raw text length of the queries often matched the gold standard's more closely than some of the other options. So, if cosine similarity rates two elements closely, perhaps comparing the lengths might lend some extra classification power.

I operationalize this intuition as follows:

- `cosineSim <- computeCosineSimilarity()`
- `textLen <- abs(queryText.length() - documentText.length())`
- `textLenFeature <- 1.0 / textLen`
- `score <- .5 * cosineSim + .5 * textLenFeature`

That is, the score is equal parts cosine similarity and a feature based on the inverse of the difference between the query text length and the current document text length. That means that documents with text lengths closer to the query's text length get higher scores, which is what we want.

With this simple change, we were able to achieve a perfect MRR score, because the new text length feature didn't dramatically change the already-correct rankings, but it did push up the score of the gold standard to be in the first spot rather than the second.

The modified system output is below:

```
score: 0.24465202695184124 rank=1     rel=true qid=1
score: 0.18554863981517647 rank=1     rel=true qid=2
score: 0.35645502494313785 rank=1     rel=true qid=3
 (MRR) Mean Reciprocal Rank ::1.0
Total time taken: 1.587
```

Note that scores in general are somewhat smaller because of the scaling factor, but the rank is preserved (and improved!) and that's what is important.

A quick note: I completed this assignment in advance of this past weekend, when additional input data was released. Since my baseline system did not have perfect MRR I decided not to use the additional data – I assume completing the original assignment as given is sufficient.