Report for PI7 – Architecture & Design Patterns

Alex Kelly – ajk1

The AbstractRanker class implements IRanker, but can itself be a Ranker or a composition of several Rankers. The CompositeRanker is handled exactly like the other Rankers, making this a good example of the composite design pattern.

In addition to the N-gram ranker which computed the cosine similarity between two texts, I wanted a ranking system that could account for minor differences between words. My other ranker looked at each token in the question and found the word that was most similar to it in the passage. I used a string edit distance algorithm here to find how many single-character alterations could be made to get from one token to the other. I then calculated a score from the sum of these token distances.

The reason I implemented this ranking method was so because the N-gram ranker distinguish “shot” and “shoot” as different words altogether. By using string edit distance I was able to recognize words with different tenses, plurality, and even spelling errors as being related. This ranker produced a slightly higher result than the n-grams did, and their composite performed even better than either.

The composite ranker simply produced a weighted average of the other scores, with all rankers weighted equally. The score threshold for a positive answer was tweaked to for the highest average F1. Below is a table of the average error analysis metrics for each ranker:

|  |  |  |  |
| --- | --- | --- | --- |
| Ranker | Avg Precision | Avg Recall | Avg F1 |
| NgramRanker | .232 | .725 | .326 |
| OtherRanker | .240 | .800 | .347 |
| CompositeRanker | .241 | .900 | .354 |