Hw3-hanz-report

Our goal of this project basically is try to use better tokenize algorithm, stemming algorithm and even better similarity measures.

For task 1, I implement basic cosine similarity, mrr.
First we use tokenze0() to get all the token vectors, and then we use cosine similarity to compute and get the ranks of each doc compared to one query. Finally we use mrr to evaluate the rank.

For task 2, when it comes to the errors, we can see that, basically it comes from 4 things.

First, there are many words that are very common in both query and doc which will highly affect the similarity, we don't want this words(which so called stop words here).

Secondly, words capitalized should be the same with their lowercased ones. So we should count the words the same if they are just differing in lowercase.

Thirdly, words with the same stem should be consider similar, we should be able to detect this.

Finally, how to compute similarity is also important, Cosine similarity may not be enough, that's why we may need BM25 which is a famous IR ranking algorithm. I also tried tf/idf to get rid of normal high frequent word and get high frequent term. I

also tried string substring kernels using a python library, which I don’t’ show in the hand in code.

What I do for task 2 are : getting rid of stop words and putting all tokens into lower cases and using Stanford Lemmatizer to get stems.