In this document, we introduce our attempt to improve retrieval performance using several different query refinement approaches.

The ones included in the code submission:

1. Pseudo Relevance Feedback: Rocchio Algorithm

With Rocchio algorithm, we want to find a query vector, that maximizes similarity with relevant documents while minimizing similarity with nonrelevant documents.

We experimented with one set of the variants called *Ide dec-hi*, which has been said to have the most effective or at least the most consistent performance.

However, the performance of this pseudo relevant feedback is not good. We tried several sets of hyper-parameters and they all give a worse score (than baseline) in the competition framework.

The reason is because, most of time, the retrieved top-k document is usually not the indeed related one. Adopting pseudo relevance feedback on wrong documents will make the result worse.

1. Zone Information

We tried to use the zone information for title part and court part of the document to improve query result by moving all document that search term in their title part and court part to the front of the result list.

However, the performance of this result is also not good (worse than baseline). We believe that is because, the test cases in competition framework do not search for a zone information directly.

1. Query Expansion:

We pre-complied a set of related term for all the terms that are known in nltk wordnet. We removed several terms using the co-occurrence matrix and added several terms using the embedding built using Reflective Random Indexing and Neural Vector Space. Some tricks are applied here to improve the competition score.

We have achieved the 1st place in leaderboard using this technique.

The ones not included in the code submission:

1. Reflective Random Indexing in paper “Discovery of Related Terms in a corpus using Reflective Random Indexing”
2. Neural Vector Space proposed by C Van Gysel

Both methods tried to build an embedding for documentation and terms so we can compute the similarity of term, query, documentation in the embedding space.

However, using them alone will not give a good result (worse than baseline). So, the embedding calculated here is just used to help refine the related terms.