query expansion techniques

**1. WordNet – Thesaurus-based**

This function is called after step1 of the search part. After tokenizing the query, we call \_expand () function to add synonyms.

Within \_expand() function, for each token, we call wordnet.synsets(token) to get a set of synonyms. In order to get more precise answers, we use a dictionary to check all synonyms. If they exist in the dictionary, add them to the query.

For example

Given the query *computer science*, after expansion, the result is *calculate computer science skill*.

**2. Relevance Feedback based on Rocchio Algorithm**

This function is called after constructing query vector.

Given a list of relevant documents and query vector:

1. for each relevant document, get the doc\_vector from postings list

2. add each doc\_vector to query vector

1. calculate the norm of doc\_vector and use beta parameter to modify it

2. add processed doc\_vector to the query\_vector

3. normalize the query\_vector by using its norm.

For example, here are our test cases:

test\_cases = [

{"query": '"Computer Science" AND Refiner can tokenize query strings into terms and tokens',

"relevant\_docs": [0, 5] },

{"query": '"Computer Science" AND Refiner can tokenize query strings into terms and tokens',

"relevant\_docs": [0] },

{"query": '"Computer Science" AND Refiner can tokenize query strings into terms and tokens',

"relevant\_docs": [5] },

{"query": '"Computer Science" AND Refiner can tokenize query strings into terms and tokens',

"relevant\_docs": []},

{"query": '"Computer Science"',

"relevant\_docs": [] }

]

"relevant\_docs": [0, 5] means the first and fifth doc are relevant documents.

A screenshot of a cell phone

Description automatically generated

Figure 1

As you can see in Fig. 1, if the relevant docs are 0 and 5, the result changes and top 2 results are 0,5. All scores of documents are also changed.