Bonus submission for HW4

# WordNet Query Expansion

Query expansion was performed by retrieving the synonyms of each query token via WordNet.

We observed that WordNet query expansion is essential for this corpus as, for example, sample q1 is "quiet phone call" but the relevant documents only have "silent telephone call" which do not match the query at all. Which lead to abysmal performance for q1:

Searching for the query "queries/q1.txt" ...

6807771 : Rank 3685

4001247 : Rank 154

3992148 : Rank 1642

With WordNet query expansion, the given relevant docs are ranked much higher:

Searching for the query "queries/q1.txt" ...

6807771 : Rank 401

4001247 : Rank 88

3992148 : Rank 392

## Effects of preprocessing on WordNet query expansion

We also experimented with 3 different preprocessing to see how well it worked with WordNet query expansion:

- Stemming

- Lemmatization (with POS set to the default NOUN)

- Lemmatization (with POS tagging)

All the results below is using WordNet query expansion.

Lemmatization with NOUN POS worked best, while Lemmatization with inferred POS tagging performed worst.

### Stemming

“nltk.PorterStemmer” is used for stemming and the following results were obtained. The WordNet query expansion is done the same way as for the below lemmatization methods, but the synonymous lemmas are further stemmed.

Searching for the query "queries/q1.txt" ...

Execution time: 0.3s

6807771 : Rank 1029

4001247 : Rank 182

3992148 : Rank 1396

Searching for the query "queries/q2.txt" ...

Execution time: 0.3s

2211154 : Rank 411

2748529 : Rank 52

Searching for the query "queries/q3.txt" ...

Execution time: 0.2s

4273155 : Rank 12

3243674 : Rank 2

2702938 : Rank 5

### Lemmatization (with POS set to the default NOUN)

“nltk.WordNetLemmatizer” is used to lemmatizing, with the POS tag given to the lemmatizer set to the default NOUN tag.

Searching for the query "queries/q1.txt" ...

Execution time: 0.3s

6807771 : Rank 401

4001247 : Rank 88

3992148 : Rank 392

Searching for the query "queries/q2.txt" ...

Execution time: 0.3s

2211154 : Rank 402

2748529 : Rank 13

Searching for the query "queries/q3.txt" ...

Execution time: 0.2s

4273155 : Rank 9

3243674 : Rank 2

2702938 : Rank 6

Lemmatization (with POS tagging)

This extends the previous “Lemmatization (with POS set to the default NOUN)” method by inferring the POS tags using “nltk.pos\_tag”.

This should, in theory, add more context to our query expansion as we now account for the word’s POS in context, but it performed the worst among the 3 preprocessing methods. A possible reason is that the inferred POS tag in the query doesn’t match that of in the document, which causes the same words to be lemmatized to different lemmas and to expand to different synonyms.

Searching for the query "queries/q1.txt" ...

Execution time: 0.4s

6807771 : Rank 3968

4001247 : Rank 106

3992148 : Rank 1258

Searching for the query "queries/q2.txt" ...

Execution time: 0.4s

2211154 : Rank 175

2748529 : Rank 141

Searching for the query "queries/q3.txt" ...

Execution time: 0.3s

4273155 : Rank 10

3243674 : Rank 2

2702938 : Rank 6

# Effects Pseudo Relevance Feedback

Pseudo relevance feedback improve our scores marginally, although the results very sensitive to the hyperparameters (alpha, beta, gamma, number of docs used). In the end, we used the hyperparameters:

alpha = 0.9

beta = 0.1

gamma = 0.1

n\_relevant = 5 (using top 5 scoring docs as relevant)

n\_irrelevant = 100 (using bottom 100 docs as irrelevant)

Locally, the ranks of the given relevant docs with and without PRF are shown below. The ranks of some documents went up, and some went down, so it was difficult to tune. It also took a lot longer to search, with q1 search time going up from 0.3s -> 5.2s.

However, it did increase our leaderboard MAF2 score from 0.319 -> 0.327.

Results from testing locally:

Without Pseudo Relevance Feedback:

Searching for the query "queries/q1.txt" ...

Execution time: 0.3s

6807771 : Rank 401

4001247 : Rank 88

3992148 : Rank 392

Searching for the query "queries/q2.txt" ...

Execution time: 0.3s

2211154 : Rank 402

2748529 : Rank 13

Searching for the query "queries/q3.txt" ...

Execution time: 0.2s

4273155 : Rank 9

3243674 : Rank 2

2702938 : Rank 6

With Pseudo Relevance Feedback:

Searching for the query "queries/q1.txt" ...

Execution time: 5.2s

6807771 : Rank 354

4001247 : Rank 91

3992148 : Rank 448

Searching for the query "queries/q2.txt" ...

Execution time: 5.0s

2211154 : Rank 278

2748529 : Rank 15

Searching for the query "queries/q3.txt" ...

Execution time: 5.3s

4273155 : Rank 8

3243674 : Rank 2

2702938 : Rank 5