Query Expansion Techniques Explored

In this project, we explored two query expansion techniques:

1. Query expansion using NLTK’s WordNet
2. Pseudo-relevance feedback

This document briefly outlines the query expansion techniques implemented, as well as how their performances are evaluated.

# NLTK’s WordNet

NLTK corpus’s WordNet is an English lexical database. It is used to expand all terms in free text queries in the following way:

|  |
| --- |
| **def** \_query\_expansion**(**self**,** list\_of\_queries**):**  """  The function expands the query  """  syns **=** **[]**  **for** query **in** list\_of\_queries**:**  lemmas **=** list**(**map**(lambda** syn**:** syn**.**lemmas**(),** wn**.**synsets**(**query**)[:**2**]))**  lemmas **=** list**(**itertools**.**chain**(\***lemmas**))**  syns **+=** **[**str**(**lemma**.**name**())** **for** lemma **in** lemmas**]**  syns **=** list**(**map**(lambda** syn**:** syn**.**replace**(**"\_"**,** " "**),** syns**))**  **return** syns |

However, WordNet is a general lexical database and not specific for the law domain. This may result in more noisy results because many of the synonyms obtained for query expansion will not be law-specific, which causes more noise.

For example:

|  |
| --- |
| >>> query = 'acquit'  >>> lemmas = list(map(lambda syn: syn.lemmas(), wn.synsets(query)[:2]))  >>> lemmas = list(itertools.chain(\*lemmas))  >>> syns = []  >>> syns += [str(lemma.name()) for lemma in lemmas]  >>> syns = list(map(lambda syn: syn.replace("\_", " "), syns))  >>> syns  ['acquit', 'assoil', 'clear', 'discharge', 'exonerate', 'exculpate', 'behave', 'acquit', 'bear', 'deport', 'conduct', 'comport', 'carry' |

Even though the word ‘acquit’ is not related to ‘conduct’, ‘carry’ or ‘bear’ in law, WordNet returns them as synonyms.

Using a law-specific thesaurus would be the most ideal.

Even so, using NLTK’s WordNet for query expansion was found to improve the MAP score of our own evaluation.

The evaluation of the performance of using NLTK’s WordNet for query expansion is done in the following way:

1. Design a query that is precisely what we want to search for. Then, get the top results and manually label them as ‘relevant’ or ‘not relevant’ based on our own judgment of the law report’s content.
2. Modify the above query by substituting the words in the query with synonyms that are less related.
3. Run the IR with the modified query without query expansion and compute the mean average precision (MAP) score of the top 10 documents retrieved.
4. Run the IR with the modified query with query expansion and compute the MAP score of the top 10 documents retrieved to see if the MAP score increases or not.

For example, a query we used for step 1 is “domestic abuse maid wife husband”. When then modify this query in step 2 to be “household aggression worker husband wife”. Without query expansion, the MAP score calculated over the top 10 results was found to be 2.3. When NLTK’s WordNet was used for query expansion, the MAP score raised to 3.2.

Therefore, query expansion with NLTK’s WordNet was deemed to have a positive impact on the search result and is included in the final submission.

# Pseudo Relevance Feedback (PRF)

Pseudo relevance feedback (PRF) is the second query expansion technique implemented in this project. For both Boolean or free text queries, a first round of search is done with query expansion. The original query vector is then modified by ‘shifting’ it slightly towards the centroid of the top five retrieved documents according to this equation

Where α is set to 0.7 and β to 0.3. This modified query vector is then used to do a second round of retrieval, which results is returned to the user.

Similar to the above, the evaluation of the performance of using pseudo for query expansion is done in the following way:

1. Design a query that is precisely what we want to search for. Then, get the top results and manually label them as ‘relevant’ or ‘not relevant’ based on our own judgment of the law report’s content.
2. Modify the above query by substituting the words in the query with synonyms that are less related.
3. Run the IR with the modified query with query expansion (previously shown to be beneficial) without pseudo relevance feedback and compute the MAP score of the top 10 documents retrieved.
4. Run the IR with the modified query with pseudo relevance feedback and compute the MAP score of the top 10 documents retrieved to see if the MAP score increases or not.

Using the same example given in the section on NLTK’s WordNet query expansion, a query we used for step 1 is “domestic abuse maid wife husband”, which is modified to “household aggression worker husband wife” in step 2. When the IR is run with query expansion but without pseudo relevance feedback, the MAP score calculated over the top 10 retrieved documents was 3.2. This value is increased to 3.3 when pseudo relevance feedback is used.

Our experiment suggests that pseudo relevance feedback mildly improves the MAP score of our IR system and would therefore be included in the final submission.