Assignment -2

Name : Chandana(18231267)

Solution 1:

Classical Rocchio Method :

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

where q_0 is the original query vector, D_r and D_{nr} are the set of known relevant and nonrelevant documents respectively, and α, β, and γ are weights attached to each term.
If we have a lot of judged documents, we would like a higher β and γ.

Starting from q_0, the new query moves some distance toward the centroid of the relevant documents and some distance away from the centroid of the nonrelevant documents.
In the classic Rocchio method we consider the feedback in a binary nature such as : Positive / Negative(1/0) feedback.
Positive feedback also turns out to be much more valuable than negative feedback, and so most IR systems set $\gamma < \beta$. Reasonable values might be $\alpha = 1$, $\beta = 0.75$, and $\gamma = 0.15$.In fact, many systems, allow only positive feedback, which is equivalent to setting $\gamma = 0$.

If we have a system where users can give feedback in a range 1 to 5. Where 1 means very non-relevant and 5 means very relevant. That means here we are dividing the Relevant and non relevant documents using this feedback score. From the given data we should control the Rocchio method How nearer the query should move to the relevant documents and how farther modified query should move wrt to Non relevant documents.

Suggestion for Modification of classical Rochio Method:

Instead of adding fixed $\beta = 0.75$, and $\gamma = 0.15$. We should tune this parameters as per the relevance order.

As per the relevancy order given by the user like :
5- Very relevant
4- Relevent
3- neutral
2- non relevant
1-very non relevant

I would like to give the weights in the range of [-1 1] :

5- Very relevant : 1.0
4- Relevent : 0.50
3- neutral : 0.0
2- non relevant : -0.5
1-very non relevant : -1.0
The weights for the documents which are not judged by the user will have the 0 weight in this method.

Example : if D1, D2 are very relevant ; D3 and D4 are non relevant

Then our modified query will be : (dot product of weights vector, and Document vector)

$$q_m = q_o + W \cdot \sum_{i=1}^{m} d$$

here our weight vector W is build based on the user feedback of each document:
W = (Very Relevant(5), Relevant(4), Not Relevant(2), Not Very Relevant(1)) = (1,0.5,-0.5,-1).
And get the modified query. Use this modified query to find the similarity. This kind of weights will move the query to nearer / farther based on the relevancy scores provided by the user.

Solution 2 :

Query Expansion Techniques:

1. Thesaurus based
2. Statistical Thesaurus
3. Spelling correction
4. Semantic Analysis

Thesaurus based query expansion :

For each term, *t*, in a query, expand the query with synonyms and related words of *t* from the thesaurus. Using **Wordnet**, Which is more detailed database of semantic relationships between English words.

Statistical Thesaurus :

1. Determine term similarity through a pre-computed statistical analysis of the complete corpus.
2. Compute association matrices which quantify term correlations in terms of how frequently they co-occur.
3. Expand queries with statistically most similar terms.

	w_1	w_2	w_3	.....	w_n
w_1	c_{11}	c_{12}	c_{13}	.....	c_{1n}
w_2	c_{21}				
w_3	c_{31}				
.	.				
.	.				
w_n	c_{n1}				

c_{ij}: Correlation factor between term i and term j

$$c_{ij} = \sum_{d_k \in D} f_{ik} \times f_{jk}$$

f_{ik} : Frequency of term i in document k

- c_{ij} corresponds to the dot product of term vectors for terms I an j on the term-document matrix

- Dot product favors more frequent terms.

Spelling correction : Spell checker can be used to find out the wrongly spelt words with the help of pre build dictionary of words using the edit distance metrics.

Psuedo feedback : Use relevance feedback methods without explicit user input.

1. Just assume the top *m* retrieved documents are relevant, and use them to reformulate the query.
2. Allows for query expansion that includes terms that are correlated with the query terms.

The above methods will give you a better query suggestions.

Solution 3 :

There are range of techniques for identifying term-term correlations from the returned collection or from the whole collection.

For Example:

- co-occurrence methods
- Distance between terms
- similarity of terms

These are very famous methods to find the correlation of words, Similarity of words

but there are some issues with these methods such as :

1. These methods doesn't give importance to the order of the sentence.
2. These methods doesn't give importance to the meaning of the words. Its Simply works based on the frequency.
3. The appropriate meaning of the word is not considered while calculating the similarity, rather it takes the best matching pair even if the meaning of the word is totally different in two distinct sentence

Suggestions for better term-term correlation score :

1. **By introducing the Similarity/ Semantics :**
    - **Word Similarity** : Before calculating the semantic similarity between words, it is essential to determine the words for comparison. We use word tokenizer and 'parts of speech tagging technique' as implemented in natural language processing toolkit.
    - **Associating a word with sense** : Every word has some synsets according to the meaning of the word in the context of a statement. Find the shortest path between the synsets. And return those words
    - **Information content of the word** : The meaning of the word differs from different domains. We can use this behavior of natural language to make the similarity measure domain-specific. Using disambiguate techniques get the sense of the word in that corpus

- **The Database – WordNet** : WordNet is a lexical semantic dictionary , we can use this for identifying the synsets. This information we can use for improving the word similarity.

By using all the above techniques we could result a better term-term scoring , This results in the better retrieval system.