Lab 5 – Query Expanding

Deadline: +1 week

1 Motivation

The performance of every search engine strongly depends on the query provided by the user. By expressing his or her question differently, different results, e.g., rankings of the most relevant pages, may be obtained. Thus, a good search engine should aid the user in (re)formulating a query. It involves, e.g., suggesting new words like synonyms, various morphological forms, or words that frequently co-occur with the words provided by the user (see Figure 1). The search engine may also fix spelling errors ("neighborhood" → "neighbourhood"; Figure 2). Some techniques are more decision-aiding oriented. For instance, the engine may allow the user to provide weights for words or ask the user to mark few documents (ir)relevant (user's feedback). Such techniques aim to improve the retrieval process and help the user find answers that are the most relevant to him or her.

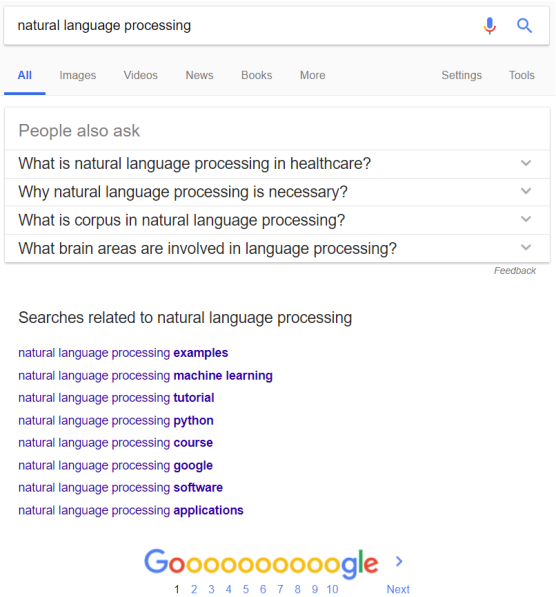

Figure 1: Google — query expansion.

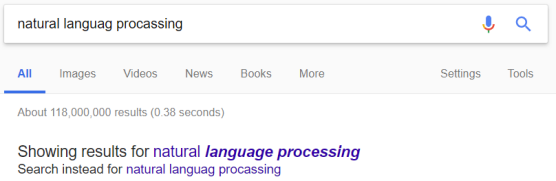

Figure 2: Google — Fixing spelling errors.

2 Query expansion based on dictionaries

Query expansion techniques can use dictionaries to suggest new words that can enrich the provided query. The search engine may use a pre-constructed dictionary that was created and is maintained by editors. Obviously, it is an expensive option. However, it is expected that the provided information has been verified and is valid (synonyms, morphological forms, etc.). An example of such a dictionary is WordNet:

- http://wordnet.princeton.edu
- a detailed database of semantic relationships between English words,
- developed by famous cognitive psychologist George Miller and a team at Princeton University,
- about 144,000 English words,
- nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called synsets.

Another example of a dictionary is Słowosieć (http://plwordnet.pwr.wroc.pl/wordnet/) -– a dictionary of the Polish language.

A dictionary may also be created automatically. It is a much cheaper and faster approach than using dictionaries maintained by the editors. However, generated data

A

	D1	D2	D3
t1="cat"	2	1	4
t2="milk"	1	0	5
t3="dog"	4	4	0

|t|
4.58
5.10
5.66

A-norm (At / |t|)

	D1	D2	D3
t1="cat"	0.44	0.22	0.87
t2="milk"	0.20	0.00	0.98
t3="dog"	0.71	0.71	0.00

A-norm-transpose

	t1="cat"	t2="milk"	t3="dog"
D1	0.44	0.20	0.71
D2	0.22	0.00	0.71
D3	0.87	0.98	0.00

C = (A-norm) x (A-norm-transpose)

	t1="cat"	t2="milk"	t3="dog"
t1="cat"	*1.00*	0.94	0.46
t2="milk"	0.94	*1.00*	0.14
t3="dog"	0.46	0.14	*1.00*

Figure 3: Correlation Matrix

may not be valid. An example of the automatically generated dictionary is a correlation matrix. Correlation matrix C is defined as $C = AA^T$, where A is a term-document matrix ($A_{t,d}$ – number of occurrences of a term t in a document d. Consider Figure 3. There are given three documents D_1-$_3$ and three terms t_1 ="cat", t_2="milk", and t_3="dog". Matrix A is a term-document matrix (e.g., "cat" occurs 2 times in D_1). A-norm is the matrix A being normalized such that every row A_t is divided by $|A_t|$. Due to the normalization $C_{i,i} = 1$. A-norm-transpose is a the A-norm matrix being transposed. See the output matrix C. You may observe that "cat" is strongly correlated with "milk". The correlation between "dog" and "cat" is medium while there is no correlation between "milk" and "dog".

3 Relevance feedback

It is not easy to provide an adequate query when the user is not acquainted with the collection of documents. It is easy to assess the relevance of a given document, though. It motivates to build a search engine based on the relevance feedback. Such a system expects that the user will provide feedback on the system's answer. In particular, the user may choose some of the documents and mark them as relevant or irrelevant. Consider the following algorithm:

1. the user provides a simple query,
2. the system returns a list of documents matching a given query,
3. the user may mark some of the documents as relevant or irrelevant,
4. the system returns the list of documents based on the automatically reformulated query.

Rocchio method a method for updating a query vector — vector space model must be used to represent documents, e.g., TF-IDF vectors. Rocchio method constructs a linear combination the following vectors:

- original query q,
- documents that are selected by the user as relevant D_r,
- documents that are selected by the user as irrelevant D_{nr}.

The new (modified) query vector q_m is constructed as follows:

$$q_m = \alpha q + \beta \frac{1}{D_r} \sum_{D_j \in D_r} d_j - \gamma \frac{1}{D_n r} \sum_{D_j \in D_{nr}} d_j,$$

where α, β and γ are weights. When the user assessed many documents, then values of β and γ should be high to increase the impact of the feedback on the elaboration of the new (modified) query. Additionally, positive feedback usually has a greater influence

than the negative ($\beta > \gamma$). Consider Figure 4. It illustrates several documents. The user selects some of them as relevant or irrelevant. The original query is depicted as well as the modified query. The weights are $\alpha = 0.5$, $\beta = 0.7$ and $\gamma = 0.1$. The β is the greatest and it can be observed that the query moved toward the centroid of relevant documents.

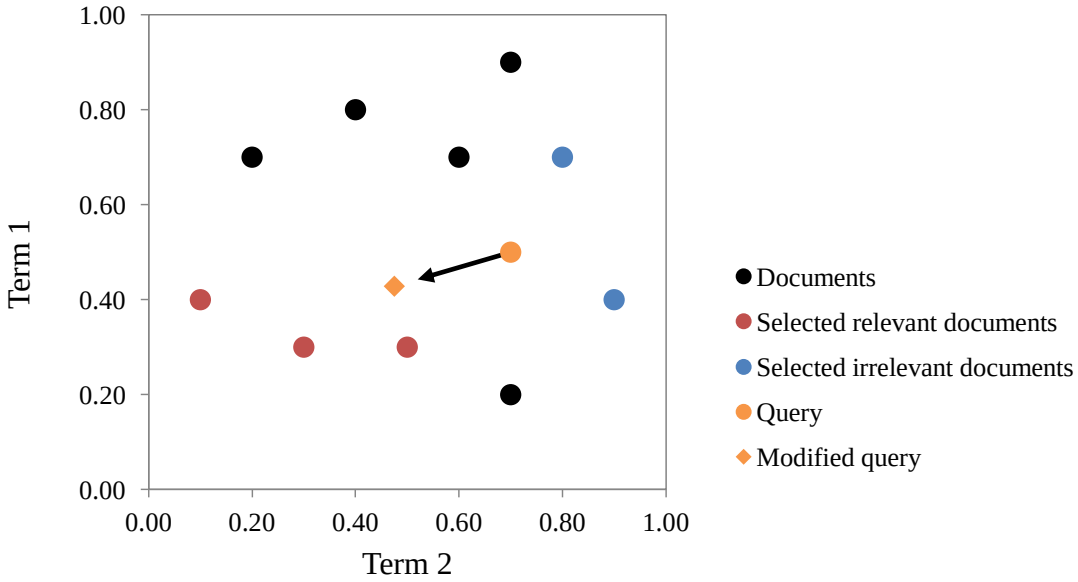

Figure 4: Rocchio method for relevance feedback: $\alpha = 0.5$, $\beta = 0.7$ and $\gamma = 0.1$.

4 Programming assignment

See the QueryExpansion.ipynb file