Web Search Project 1

by Steven Xu, 350256

1 Database Creation and Storage

Data was stored using python's *pickle*, as the *shelve* module was too slow to be used on windows, which was the primary development platform.

Only the frequency matrix was stored instead of the tf-idf matrix for flexibility, as the frequency matrix was needed to be used for later sections. The tf-idf matrix was then calculated upon initialization time when search queries were run.

The frequency matrix was stored as a sparse row matrix. A sparse column matrix was not used as rows were accessed more often than columns.

The inverted index was stored as a python dictionary, with each term mapping to a list of document id's.

2 TF-IDF Results

See results.txt.

3 Pivoted Length Normalization Results

Increasing s increeases the effect of unit-length normalization, which decreases the gain in weight long documents have. When $s = 1$, all term vectors (documents) are normalized to the same length, which removes the effect of pivoted length normalization.

See results.txt.

4 Requiring All Query Terms

See results.txt

5 Disambiguation by Source Context

5.1 Rocchio's

We can perform query expansion using the user's current forum post and the subforum he's in. Let q_0 be the original query, d_u be the user's current post and D_F be the set of posts in the subforum, then we use the expanded query vector:

$$q_e = \alpha q_0 + \beta d_u + \gamma \frac{1}{|D_F|} \sum_{d \in D_F} d$$

For efficiency we then take the top k terms which are common to the wikipedia articles.

5.1.1 Forum Weights

Tf-idf weights for forum posts perform poorly, as they can down-weight terms like "apache", which are common in the forums but not the encyclopedia.

Instead, the following weighting scheme was used which tries to extract common terms in the forum but not in wikipedia:

$$w_t = \log\left(1 + \frac{1}{|T_A|}\sum_{d \in D_A} f_{d,t}\right) - \log\left(1 + \frac{1}{|T_W|}\sum_{d \in D_W} f_{d,t}\right)$$

Where D_A is the set of apache forum posts and D_W the set of wikipedia documents, and T_A, T_W the set of terms.

5.2 Rocchio's with SVD

We can use still rocchio's, but first transform the set of wikipedia documents and apache forum documents into a semantic space using SVD. We include the forum documents to extract more relevant topics.

We can then use Rocchio's, then instead of using term weights, we are using topic weights, which might provide more query expansion. This gives more importance to topics contained in the subforum and in the post.

5.3 Cluster Distances

We can interpret the forum documents as a hierarchical cluster, and give a bonus weight to wikipedia documents closer to the clusters of forum documents that we are looking for. Let d be a distance function between a document and a set of documents, then we want a document, r which minimizes:

$$\alpha \cdot d(r, \{q_0\}) + \beta \cdot d(r, \{d_u\}) + \gamma \cdot d(r, D_f)$$

If we take the average distance, $d(r, D) = \frac{1}{|D|}\sum_{d \in D} r \cdot d$, then we get rocchio's query expansion. But we can also take the minimum distance to the cluster, using $d(r, D) = \min \sum_{d \in D} r \cdot d$.

6 Disambiguation Implementation

See results.txt.