**Lab 2: Search Engine Competition**

Alexandru MARINESCU

In this assignment I will test different rankers that are build-in in MeTA and tune their parameters with the purpose of obtaining better results in searching the MOOC's dataset. I will also test different combinations of analyzers and filters to try to obtain better MAP results.

To test my implementation I ran “competition”, a program that receives as a parameter a configuration file containing all the necessary detail about the files containing lists of stop-words, prefixes, punctuation signs and many others. The configuration file also contains information about which ranker function should be used alongside with it's necessary parameters (e.g. dirichlet-prior with it's tunable 'mu' parameter). The analyzers, tokenizers and filters are also set in the configuration file. In the beginning I will use the default analyzer having method="ngram-word" and ngram=1 which means that the tokenizer will segment each document into unigrams. The default filter will actually perform multiple predefined filters such as lowercase conversion, length filtering and stemming.

First I ran the test with the following rankers, without tuning the parameters: bm25, dirichlet-prior, jelinek-mercer, pivoted-length and absolute-discount and observed the MAP.

To tune the parameters of every ranker I implemented a bash script that would allow me to tune every ranker independently by dynamically editing the configuration file to change the values of the parameters. The script also allows me to run the test with all the ranking functions to determine which one gives the highest score(MAP).

The actual tuning process will be exhaustive, trying different values within an interval and observing which one leads to a higher MAP value. All the intermediary values will be stored in files named after the rankers. The maximum MAP values and the parameter configurations that lead to them will also be stored in a separate file, in order to compare the rankers among themselves.

A better tuning approach for most ranking functions would be to start with a low value for the configuration parameter and a high incrementation step and test the it to obtain the MAP values. Once a decrease in the MAP values is detected, the incrementation step should be lowered and negated in sign(i.e. start decreasing the parameter value). This step can be performed multiple times leading to a better value for the configuration parameters and consequently a better MAP result.

By simply tuning the ranker's parameters I reached the following MAP results:

dirichlet-prior: mu = 325 ; MAP = 0.51051

jelinek-mercer: lambda = 0.70 ; MAP = 0.509682

pivoted-length: s = 0.15 ; MAP = 0.502367

b25: b = 0.8 ; k1 = 1.8 ; MAP = 0.507677

absolute-discount: delta = 0.65 ; MAP = 0.506567

To further improve my results, I used a combination of analyzers and filters and after modifying the configuration file to include them, I reran “index” with the beforementioned configuration file as a parameter to reindex the mooc's database. I first used a customized filter chain composed of an icu-tokenizer, conversion to lowercase and imposed a length limit (2 to 25 characters), but the results obtained were lower than the ones obtained by using the default MeTA filter(default-chain). This was somehow predictable because the default filter chain also performs the same filters with an additional stemming filter. I then used a combination of two analyzers in combination with two of MeTA's default filter chains configuring the text analysis process to consider unigrams and bigrams of words generated by running the documents through the two filter chains: “default-unigram-chain” and “default-chain”.

I obtained the following results:

dirichlet-prior: mu = 760 ; MAP = 0.511753

jelinek-mercer: lambda = 0.70 ; MAP = 0.509682

pivoted-length: s = 0.15 ; MAP = 0.502367

b25: b = 0.95 ; k1 = 1.85 ; MAP = 0.513496

absolute-discount: delta = 0.75 ; MAP = 0.487791

In the last step I tampered with the file containing the list of stop-words. I initially removed random rows from the file and observed a decrease in the MAP score. I then browsed the internet using a real search engine and found a list containing around 200 more english stopwords. I indext the MOOC's dataset again after modifying the configuration file to include the new list of stopwords. I reran the test and observed the new MAP values. I then concatenated the original list of stopwords with the new one and repeated the process (index and test).

I obtained the following results:

dirichlet-prior: mu = 760 ; MAP = 0.513928

jelinek-mercer: lambda = 0.70 ; MAP = 0.512219

pivoted-length: s = 0.15 ; MAP = 0.503013

b25: b = 0.95 ; k1 = 1.85 ; MAP = 0.51416

absolute-discount: delta = 0.75 ; MAP = 0.491252

Relevant resources:

https://github.com/alexmar95/MeTASearchEngineTuning/