Search Engine Peer Review

Crawler

Intro

It's nice that he has threads in his crawler and that he made a GUI for it.

Where is querying done? It's not apparent in the image of the GUI.

Url Frontier

It's nice that he has some way of ensuring that hosts are not hit all the time, instead of just waiting some set amount of time after each hit.

Could have used the Mercator Scheme instead, as that might work better.

Retrieving links

Links are converted to lowercase. Urls are case-sensitive according to W3. Lowercasing could give problems.

Might not be entirely breadth-first because getting the next page might throw Uri's that have already been visited to the back of the queue.

Politeness using robots.txt

Might also be able to look at Allows, but that's an extension and only Disallows really exists so that might be okay.

How often is this check run? It's not completely obvious when it's used.

Removing duplicates and near-duplicates

Why not implement this? Is it just a call to the GetJaccardSimilarity method that you would need to add?

Shingling could be cached as it is fairly heavy.

Indexer

Intro

It's not strictly necessary to use HtmlAgilityPack here, could just remove all the HTML tags and get the text content that way.

However it might be an advantage to use HtmlAgilityPack here though because you could possibly catch more text content, given meta tags. Also more clear what it is doing than for example using Regular Expressions to just get rid of all the tags.

Trimming and case folding

OK, might want to go into a bit more detail about why you do this.

Stop words

OK, might want to go into a bit more detail about why you do this.

Symbols

Why are they removed?

Stemming

We used a library that implements Porter's Steming Algorithm,
see http://tartarus.org/martin/PorterStemmer/csharp.txt

Frequency Index

Is it token frequency or term frequency?

Very neat and concise implementation of the frequency index.

Inverted Index (Boolean)

Again very neat and concise.

Ranker (content based)

Intro

OK.

Ranking

"inversed index" is a typo, fix to "inverted index"? Try to explain what tf-idf weighting is in a bit more detail.

Querying

The query needs to be treated the same way as the documents when they are indexed, so it might be a good idea to apply these normalizations to it.

Overall

Overall the architecture of the search engine is good because it makes it clear what everything is doing. The more complicated parts are written in clear and concise ways so that is also good.

A spammer could put words into the background to make it higher ranked with regards to almost any query.