FollowMyLink

Individual APT Presentation

Third Talk

February 2009

Overview

- The Page Model
- Adding Terms to the User Model
  - Stemming
  - Determining the significance of terms
  - Catering for the term original!
- Extracting a query from the User Model
- Selecting a FollowMyLink destination

The Page Model

- You’ll need, at some point, to represent the information present on a page if you want to figure out which bits of text are significant by their position in the document
- I assume that you will only process HTML or its derivatives
  - Possibly convert pdf/doc to text/HTML, but i) text may lose information about structure, ii) HTML may create enormous number of tags

The Page Model

- Algorithms:
  - HyperContext - may need to be modified
    - http://www.cs.um.edu.mt/~cstaff/HCT/thesis/hct7.pdf Section 8.3.2
  - VIPS - may need to be rewritten to make it platform independent
    - http://research.microsoft.com/research/pubs/view.aspx?tr_id=690
  - DOM - just needs to be accessed once the user has made their selection or clicked an existing link to follow, but write your own algorithm!
    - http://www.mozilla.org/docs/dom/domref/dom_intro.html

The Page Model

- If you use DOM, then just because you have found a block of text that the user has clicked from, does not mean that you won't need to use a topic segmenter!
- DOM will return smallest enclosing block of text (region)
  - But region may be large!
  - Region may be related (similar) to other regions
    - E.g., header, title, other (possibly adjacent) blocks of text

Finding *Context Blocks*

- See HyperContext Section 8.3.2 for a simple description
- You can either make assumptions about the structure of information in a page, or you can compare the selected region to others regions to measure similarity, merging the most similar regions
- You may wish to segment large regions first!

Comparing Regions

- Either implement your own matching algorithm
- Or use a third party one
  - E.g., a document indexer and search engine!

Steps in Matching

- Stem terms
  - Use e.g., Porter's Algorithm or similar
- Remove stop words
  - Find a good stop list (or stop word list)
- Index the remaining stems
  - Using e.g., term frequency x document (region) frequency to calculate term weight
  - Can also add weight to reflect positional info in doc

Steps in Matching

- Convert the region containing the link the user followed or from which the user requested FollowMyLink into query
- Use e.g., Cosine Similarity Measure to compare ‘query’ to representations of other regions
  - Remember not to index region representing query!
- Use terms in highest ranked region, or terms in regions that rank higher than some threshold

Third Party Search Engines

- SMART
  - http://en.wikipedia.org/wiki/SMART_Information_Retrieval_System
  - Tutorial @ http://www.tcnj.edu/~mmmartin/CSC485IMME321/Papers/SMART/SmartCourse.html
- Google Desktop
- Lucene
- SWISH-E
- Also see http://www.webir.org/resources.html
- You want it to be lightweight and portable, because it will be installed on user machine!!!

Adding terms to the User Model

- Once you’ve found the relevant regions, you can add the terms (stems) to the user model
- THIS MEANS THAT THE SEARCH ENGINE MUST ALLOW YOU TO EXTRACT TOP RANKING TERMS FROM THE INDEX FOR THE REGIONS!
- Probably as hard to use 3rd party engine and integrate it as it is to write you own simple one!

Adding terms to the User Model

- Age the terms in the User Model before updating it
- Remember to also include the different variants of a word (before they are stemmed) so you can either OR them or select the most frequently occurring to submit to the Web-based search engine to select a destination for FollowMyLink

What do you do if…?

- The user follows a bookmark/favourite, rather than a link?
- The user follows a link whose source is an image?
- The user types an address directly?
- Goes to a destination via a search page?
- Uses some other mechanism to go to a page?

Extracting a Query from the UM

- So, you’ve got a user model, and your user has followed a FollowMyLink…
- We need to extract a query from the user model to submit to one (or more!) Web-based search engines
- Take the most significant *n* stems, and submit them using them in their original form

What to do next

- Next, devise a strategy for selecting the destination of the FollowMyLink
- Remember, the user *may* wish to see all the results
- Remember, the user *may* wish to modify the query (should this result in an updated UM???)
- Remember, the page the user FollowedMyLink from *may* be in the results page!

What to do next?

- If you poll several search engines, you *may* wish to select the most frequently occurring destination

That’s it!!