Analysis of News Articles using Meta Search Engines

Project Report

Submitted by:
Vivek R. Shivaprabhu
vivekrs.rsv@gmail.com

September 2008

Contents

Introduction....3
Purpose....3
Approach....3
PART A: Identifying the News Search Engines....4
PART B: Building the Meta Search Engines....6
PART C: Analyze and Segregate Results....6
Retrieve a set of articles from the Meta Search engines C and L....7
Retrieve x top articles from C and another x top articles from L....8
From the 2x articles, identify top z words which are prevalent in L but not C and another top z words which are prevalent in C but not in L....11
Retrieve sentences containing these words and separate them into 2 columns....14
Grouping sentences and terms....16
Words occurring frequently on both sides....16
Determining the Sentiment....16
Running the application....17
Other works of interested....17
Works Cited....17

Introduction

The internet hosts innumerable news outlets that cover events worldwide. These providers provide search engines to help users query for news. A Meta Search engine submits user-supplied queries to several search engines simultaneously and retrieves results obtained from their databases without using a database of its own (1). News Meta Search engines aggregate several news sources thus providing results from several news sources in one view.

Ideally, a news source should report events and information without making any judgments or without imposing the writer's views on the reader. Clearly, since journalists are human beings, and since news organizations essentially follow the agenda of its owner, every news source is biased. When a news search engine pulls information from liberal sources, the articles retrieved will seem one-sided. Similarly, if a user queries a conservative search engine, the news articles will seem conventional and resistant to change.

It would be nice to see what kinds of words are used more in conservative news articles and what words are used in liberal articles.

Purpose

For any given query, we wish to find the critical differences between the liberals and the conservatives. By repeating this exercise for myriad queries, we will be able to identify the liberal and conservative stand-point on various issues. The issues differentiating the liberals from the conservatives should be identified.

Approach

a. Identify a set of liberal news search engines, and a set of conservative news search engines, say 10 to 15 each.

b. Build two Meta Search Engines – L, which connects to the liberal search engines; and C, which connects to the conservative search engines.

c. For any submitted query, retrieve a set of links to articles from C and L.

d. Retrieve x top articles from C and another x top articles from L.

e. From these 2x articles, identify top z words which are prevalent in L but not C and vice versa.

f. Retrieve the sentences containing these words and separate them into 2 columns.

PART A: Identifying the News Search Engines

News search engines can be broadly classified as liberal or conservative (or neutral) based on self-identification or the apparent content. The following tables list some search engines in both categories:

Table 1: Conservative News Sources

News Source, Hyperlink	Reason for classification as Conservative
Free Republic - http://www.freerepublic.com/home.htm	Self Description – "Free Republic is the premier online gathering place for independent, grass-roots conservatism on the web."
The Conservative voice - http://www.theconservativevoice.com	As the name suggests.
Media research centre - http://www.mrc.org/about/aboutwelcome.asp	http://www.mrc.org/about/aboutwelcome.asp - "Leaders of America's conservative movement"
Cybercast News Service - http://www.cnsnews.com	As claimed by Wikipedia - http://en.wikipedia.org/wiki/Cybercast_News_Service
Federal Observer - http://www.federalobserver.com/faq.php	As claimed by http://www.sourcewatch.org/index.php?title=Conservative_news_outlets
Right Wing News - http://www.rightwingnews.com	As explained in their FAQ - http://www.rightwingnews.com/faq.php
Fox News - www.foxnews.com	As claimed by Wikipedia - http://en.wikipedia.org/wiki/Fox_News_Channel
American Conservative Daily http://www.americanconservativedaily.com	As the name suggests.
Pardon My English - http://www.pardonmyenglish.com	Self Description – "Conservative news & opinion."
Conservapedia - http://www.conservapedia.com	As the name suggests.
National Public Radio - http://www.npr.org	As claimed at http:// en.wikipedia.org/wiki/National_Public_Radio
National Review - http://www.nationalreview.com	As claimed by Wikipedia - http://en.wikipedia.org/wiki/National_Review
The Land of the free - http://www.thelandofthefree.net/index.php	Self Description on website
The new media journal - http://www.therant.us	As explained at http://www.associatedcontent.com/article/94838/best_conservative_politics_fan_sites.html
Newsmax - http://newsmax.com	As claimed by Wikipedia - http://en.wikipedia.org/wiki/Newsmax

World Net Daily - http://worldnetdaily.com	As claimed by Wikipedia - http://en.wikipedia.org/wiki/Worldnetdaily
The Washington Times - http://www.washingtontimes.com	As claimed by Wikipedia - http://en.wikipedia.org/wiki/Washington_Times#Relationship_to_the_Unification_Church

Table 2: Liberal News Sources

News Source, Hyperlink	Reason for classification as Liberal
Common Dreams - http://www.commondreams.org	Calls itself a part of progressive community in its self description
TomPaine.com - http://tompaine.com	Calls itself "The best progressive insight and action".
MotherJones .com - http://www.motherjones.com	As explained in Wikipedia - http://en.wikipedia.org/wiki/Mother_Jones_%28magazine%29
The Nation - http://www.thenation.com	As explained in Wikipedia - http://en.wikipedia.org/wiki/The_Nation
Alternet - http://www.alternet.org	As explained in Wikipedia - http://en.wikipedia.org/wiki/AlterNet
The New Republic - http://www.tnr.com	Calls itself "Liberal news outlet"
New York Times - http://www.nytimes.com	As explained at - http://www.sourcewatch.org/index.php?title=New_York_Times
The Raw Story - http://www.rawstory.com	Called left-leaning at http://en.wikipedia.org/wiki/Raw_story
OpedNews.com - http://www.opednews.com	As its slogan suggests
Atlanta Progressive news - http://www.atlantaprogressivenews.com/news.html	As the name suggests
My Antiwar.org http://www.myantiwar.org	As the slogan suggests.
Liberal news topix - http://www.topix.com/city/liberal-ks	As the name suggests
Huffington Post - http://www.huffingtonpost.com	As mentioned at - http://en.wikipedia.org/wiki/Huffington_Post
Truthdig - http://www.truthdig.com	As mentioned at - http://en.wikipedia.org/wiki/Truthdig
CNN - http://us.cnn.com	As claimed at - http://en.wikipedia.org/wiki/Cnn
MSNBC - http://www.msnbc.msn.com	As claimed at - http://en.wikipedia.org/wiki/Msnbc

PART B: Building the Meta Search Engines

Suppose each of the news search engines returns a set of articles such that each article contains the attributes {DateTime, NewsHeadline, ArticleBody, URL}. A Meta Search engine, C, is built which aggregates news articles from all the conservative news search engines. Another Meta Search engine, L, is built for the liberal news search engines.

For a given query, C and L will combine the various news search engines in its respective category and return articles containing the attributes {NewsSourceName, DateTime, NewsHeadline, ArticleBody, URL}.

Finally, a Meta Search engine, C and L, connects the results from C and L. A query to C and L will return articles from C and L that contain the values for the following attributes: Each record of the result-set may contain the following attributes:

Table 3: Possible attributes from the news Meta Search engine, C and L

Field	Description
[C|L]	The news Meta Search engine (Conservative or Liberal) from which the article was retrieved.
NewsSourceName	The name of the news paper/source that supplied the record to C or L.
DateTime	The timestamp associated with the creation date/time of the article.
NewsHeadline	The headline of the article.
URL	The URL where the article can be read.
ArticleBody	The body text of the article.

My Search View (2) is a customized Meta Search engine generator which connects various custom-defined search engines and merges the data extracted from the results from each search engine (3).

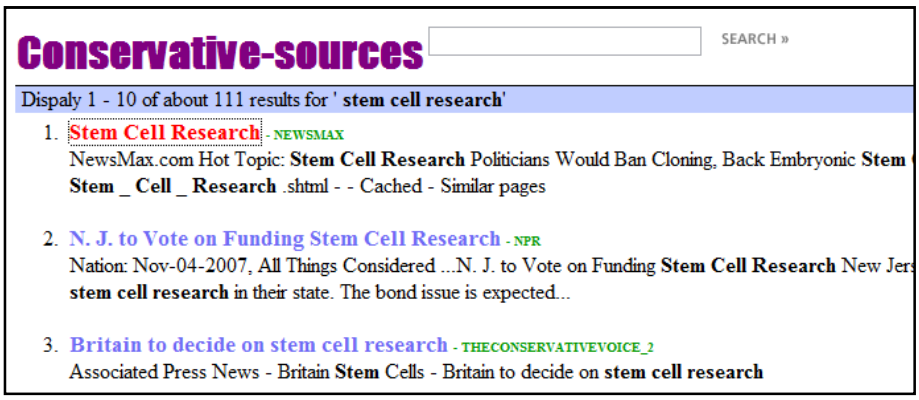
Google Co-op provides an interface that can be used to deliver specialized results from specific sites. Since this is based on crawler technology, it might not be able to give up-to-date results.

PART C: Analyze and Segregate Results

The system consists of two Meta Search Engines and a text analysis system which in turn uses The Porter Stemming Algorithm, Word Net, The Stanford Part-of-Speech Tagger and other indigenous tools to find characteristics in the texts. The text analysis system is built using Java and produces HTML pages containing the results of the process. The steps involved in analyzing and segregating articles are as follows:

Retrieve a set of articles from the Meta Search engines C and L

Using the Meta Search engine, a user query will return articles that satisfy user criteria. Two Meta Search engines have been created using MySearchView – Conservative-sources and Liberal-sources. These two search engines extract data from some of the sites mentioned in Part A in their respective categories. A sample results page for a query ran on the Meta Search engine would like this:



The way in which results are displayed in this application has been simplified so that the data can be easily carried forward. The query is specified through a URL as follows:

```
http://<server_path>?
      mse=[conservative|liberal]&
      q=<query_string>&
      d=<show_date>
```

where

mse : the Meta Search Engine to query

q : the actual query to send to the Meta Search Engine

d : 'y' if the publish date for the articles is to be fetched

The page returned by the Meta Search Engine has a simple body containing all the search results in a single page, with each record encapsulated within a list item of an ordered list.

The followings fields are displayed, delimited by new line (`<br/>`):

Table 4: A record in the results page of MySearchView

Field	Description
URL	The URL where the article can be read.
NewsHeadline	The headline of the article (= the title of the page).
ArticleSnippet	A snippet of the body text of the article containing the search query.
ArticlePublishDate	The date time that the article was published on.

From this data, the required fields can be easily derived. We are currently able to extract the following data:

Table 5: Data extracted from MySearchView

Field	Remarks
[C|L]	This can be obtained easily as of now, as we are directly searching for Conservative sites or Liberal sites.
NewsSourceName	This is extracted from the URL. Everything after http:// up to the occurrence of the first slash (/) is the source name.
DateTime	This is extracted from the ArticlePublishDate field.
NewsHeadline	Equal to the NewsHeadline field in the results.
URL	Equal to the URL field in the results.
ArticleBody	The article body is extracted by going to the web page pointed to by the link in the result.

Retrieve x top articles from C and another x top articles from L

From the results obtained from C and L, a set of top x articles are to be chosen. The criteria for selecting the top articles can differ depending upon the data that is available from C and L. The Meta Search Engine sorts the articles based on relevance and date. Some articles from C will be more strongly conservative than others. It will be interesting to pick these articles than those which are not-so-conservative. The same idea applies to liberal articles. This is hard to identify. Therefore, the more articles we get, the better it will be to compare and contrast the results. By default, 30 articles are fetched from C and 30 from L. This can be modified by the user by choosing another number at the prompt when the program is executed.

A check is made to ensure that the program does not download large PDF files. If a link seems to be a PDF, it is ignored.

The URLs are reordered such that pages from different web sites are retrieved. If there is more than one URL from the same web site in the top results, the second and subsequent URLs from the same website are pushed down. Consider a scenario in which 15 pages were returned by the metasearch engine and 10 of them have to be retrieved. The following table illustrates the reordering of results: (a page is represented by a character which indicates the website it belongs to and a number which indicates the

page number of the page in the website. Example: B3 represents the third page from website B.)

Order	Page returned by MSE	Page after re-ordering
01	A1	A1
02	A2	B1
03	A3	C1
04	B1	D1
05	C1	E1
06	B2	A2
07	A4	B2
08	C2	C2
09	B3	D2
10	A5	A3
11	A6	B3
12	D1	A4
13	D2	B4
14	B4	A5
15	E1	A6

Using this method, a greater variety of articles is fetched from different domains, than similar articles from the same website being fetched as top news articles.

Some web pages redirect to Wikipedia articles. Such pages state facts rather than give the author's opinion about the issue or event, which we are not interested in. However, the URL for the page contain the domain name of the news provider. By looking at the `title` tag of the page, we can determine if the page is from Wikipedia. If it is, the page is ignored and the next page is fetched from the results of the metasearch engine.

The text of a news article will contain several quotes by people that are enclosed within quotation marks. There are several types of Unicode Quotes that are represented using special codes in HTML (example: `&quot;`, `&amp;`, etc). Such codes are replaced by their corresponding character (example: ", &, etc).

A news article contains several paragraphs of text, each paragraph containing several sentences. Sentences are delimited using the symbols period (.), colon (:), exclamation (!) and question mark (?). Other rules are also incorporated: Very small sentences are not usually very useful. They tend to contain copyright notices, links to more information, etc. In order to avoid such sentences, only sentences containing more than ten words that are of use (a word is considered useful if its Part of Speech is Noun, Verb, Adverb or Adjective) are used. If a sentence contained a short clause like "`John Doe wrote:`" or "`All Rights Reserved`" or similar frequently occurring sentence, it is removed. Hyperlinks are a special case. Since they contain both colon (:) and periods (.) such 'sentences' are neglected.

Sometimes, there are words (or phrases) within brackets separated by one of the sentence delimiters. In such a case, the remaining part of the sentence within parenthesis will show up as the start of a new sentence. For example, if there is a reference number containing a period the words prior to the dot become one sentence and the words after the dot become another. This is undesirable and handled as a special case. While parsing the sentence, if the current sentence has fewer than three words and a closing bracket is encountered, then the initial portion of the sentence is eliminated.

The article body is a set of one or more paragraphs of text. Typically, a news page will contain text along with links to related and similar articles, images, advertisements, etc. It is a challenge to retrieve just the text pertaining to the article and text pertaining to the user query. Only the text inside the `<p> … </p>` tags are extracted. All tags are removed from the underlying HTML. Any text within the `<script> … </script>` tags is removed.

It is possible that a long article contains text/information about more than just the user query. For example, a query on *stem cell research* on the Meta Search Engine may return articles that discuss the opinions of politicians on various current issues. However, we are only interested in the part of the article concerning the actual query. Therefore, only paragraphs containing one or more of the query words are extracted from the article. This will filter out texts not related the query to some extent. This has greatly improved the quality of the resulting sentences. Sometimes, the entire page may not have the query terms in it (as a result of some kind of an error in the search). In such a case, the article text will be blank. When the search engine returns a page, the page may not contain all the query terms close to each other. Paragraphs that contained at least 50% of the query terms were being retrieved. It was found that some of the sentences containing significant words were not relevant. This rule is now modified to pull out paragraphs that contain all the query terms in it. As a result, the number of web pages rejected has increased, but the relevance of the significant word/phrases has increased.

It is observed that some of the news providers first display a full-screen flash advertisement before going to the actual article. There are also pages in some popular news sites that require a user to login to view the article. In order to extract the text article, either a button has to be clicked on the page, or a valid user's credentials have to be entered, which will result in a new page to be fetched. This will vary depending on the article and the website, hence could not be handled by code. In such cases, the actual text for the article will be blank, although there is a lot of HTML and other information in the page. Such pages are ignored.

A check is made to see if there is any valuable text after parsing the entire page. If there is no text as a result of no paragraphs being retrieved or because of the flash

advertisements, the entire page is discarded and an additional web page will have to be fetched. The user is notified in such a case through the debugging messages.

Most news pages have User Comments following the article body. These are comments submitted by the people who read the article. Although some of these comments may be useful, most express the opinion of the individual user and may not be along the same ideology as the news provider. Some comments are derogatory and not useful. Therefore, it is best if we do not parse these paragraphs. At the time of parsing the page, when the word `comment` appears inside a `heading` tag (Example: `<h2> 52 Comments: </h2>`), the rest of the page is not read.

There have been instances where an article is replicated on more than one URL. This could be due to several reasons: the two URLs are articles from different news providers who are reporting the exact same news item, which is issued by another source; the two URLs could be two versions of the same article in the same news site – one with images, and one without. This is handled by checking the sentences read by the articles. If a sentence parsed from an article is present in another article that was read, then the article with the lower rank is discarded. Debugging information will notify when such a situation occurs.

POS tagging and stemming of words (which are discussed in greater detail in the next sub-section) in a lot of articles is time consuming. Moreover, users will not be interested in seeing a lot of sentences as results for a given query. Choosing a good number of articles to process is a decision to be made by the user (technically, the program can process as many articles as required by the user). Currently, a default value of 30 pages each for conservative and liberal articles are read and sentences for 20 most significant words are shown.

From the 2x articles, identify top z words which are prevalent in L but not C and another top z words which are prevalent in C but not in L

We start with the assumption that there are certain words or word-pairs that occur more frequently in liberal news articles and other words (or word-pairs) that occur more frequently in conservative news items. Single words as well as pairs of words (example: `ceramics collection`) are chosen to determine frequencies.

The Part-of-Speech of the words in articles from C is identified using a Part-of-Speech tagger (4). Nouns, Adjectives, Adverbs and Verbs are picked as the interested words.

The words in the user query are taken as granted irrespective of whether they satisfy any of the above conditions. This way, we may be able to generate word-pairs that contain a word in the query, which would be highly desirable. This ends up getting a high ranking for query words in the result. Therefore, if a word or word-pair is a direct substring of the query string, it is ignored (not displayed) when the results are generated. For example, if the query string is `stem cell research`, word pairs like

`embryonic stem` may be interesting to see, but words pairs like `stem cell` are not.

POS tagging eliminates stop words (`and, because, their,` etc) up to a level, but not completely. Therefore, a list of common stop words are loaded at startup and stored in a HashMap. This list is used to eliminate stop words to a greater extent (Example: `copyright, write, said,` etc).

Each word (and word pair) extracted is reduced to its stem, or root form using Stemming, also called Conflation (5). (6) has a good implementation of the original Porter stemmer for English, which is used to obtain the root of the inflected or derived words.

There are words that have different in stems, but are used as synonyms (example: `murder`, `slay`, `execute` are words that have the same sense). It will be interesting to group such words. By doing so, similar words will contribute to each other and thus improve their rank as a whole. For every word that is read, (7) is used to find synsets. These word forms are grouped together as one WordGroup object. Therefore, when calculating chi-square values, the term 'word' actually refers to a group of words that are similar, i.e., synonyms, derivationally related forms and words that contain the same stem. Only if the word is a Noun or a Verb, it is checked for similar words, because adjectives and adverbs have many unrelated meanings for the same word. However, even nouns and verbs may have more than one sense associated with it (example: the noun `issue` has more than ten senses associated with it, and each sense has many synonyms and derivationally related forms). The frequency of use of the words is used to determine if two synonymous words (or pairs) can be grouped together. Suppose a word `w` has two or more senses. These senses are arranged in the descending order of their frequency of use. Each sense will have several words (or phrases) as synonyms and each word (or pair) will have a number associated with it which indicates its frequency of use. Consider a synonym word w_1 in a sense. It can be used as a synonymous word to the given word `w` if the following condition is satisfied:

$$\frac{frequency\ of\ use\ of\ w1\ in\ w}{sum\ of\ all\ frequencies\ of\ all\ senses\ of\ w} * \frac{frequency\ of\ use\ of\ w\ in\ w1}{sum\ of\ all\ frequencies\ of\ all\ senses\ of\ w1} \geq 0.5$$

Different forms of some words will result in different stems. For example, `embryonic` will stem to `embryon`, whereas `embryos` will stem to `embryo`. Ideally, these two words should have been grouped together into a single Word Group. But since they become separate Word Groups, they will not come up as significant terms. This is undesirable, since they are essentially the same thing. To overcome this, derivationally related forms of the words are taken from WordNet and compared to existing Word Groups. If they are found to match, such word groups are merged together. In our

example, `embryonic`, `embryo`, `embryologic` will form one word group. This results in higher chi-square values for significant terms.

Capitalized words are a special case. For example, the noun 'States' is different from the verb 'states'. Therefore, such capitalized words are stored as-is.

While parsing the sentences, if the current character is a comma, we are not interested in the next word forming a word-pair with the current word (example: consider the sentence: `These two leaders share much common ground, particularly in opposing abortion, embryonic stem cell research and gay marriage.` Here, the word-pair `abortion embryonic` may occur frequently, but is of no use to us. Such word-pairs are removed from the list of frequent words.

While we are at it, we also separate out the sentences in the articles for use in the next step. They are separated based on period (.), exclamation (!) and question mark (?) symbols. Special cases for period being used in acronyms have been written. So, a sentence like "`They made the comments in support of the Patients First Act of 2007 (i.e., a cure for disease) at a news conference in Washington, D.C. late last week.`" will be pulled out correctly in spite of words like `D.C.` and `i.e.` contain dots in them. Also, conditions to take care of words like `Prof.`, `Ph.D.`, etc have also been included. To achieve this, the length of the last word in the sentence is measured. If the word has only one character, it is definitely a part of an acronym. It is also an acronym if it has four or fewer letters with the first letter capitalized.

For every stem word (or word pair) `w`, a count of the number of occurrences in Liberal and Conservative sentences is stored. Then, the chi-square (8) is calculated to verify the assumption that w is unevenly distributed in the results from the two search engines C and L.

Table 6: A chi-square test

	No. of sentences containing w	**No. of sentences not containing w**	**Row total**
Conservative articles	w_c	w_c'	$\boldsymbol{w_c + w_c'}$
Liberal articles	w_l	w_l'	$\boldsymbol{w_l + w_l'}$
Column Total	$\boldsymbol{w_c + w_l}$	$\boldsymbol{w_c' + w_l'}$	$\boldsymbol{w_c + w_c' + w_l + w_l'}$

The number of sentences containing w and the number of sentences not containing w in the articles from the conservative Meta Search engine and the liberal Meta Search engine (w_c, w_c', w_l, w_l') are observed. For each observed value O, the corresponding expected value C and L is calculated using:

$$E = \frac{row_{total} \times column_{total}}{grand_{total}}$$

For example, the expected value C and L for the observed value O=w_c is calculated using:

$$E = \frac{(w_c + w_c') \times (w_c + w_l)}{(w_c + w_c' + w_l + w_l')}$$

The chi-square value for this observed value is calculated using:

$$\chi^2(w_i) = \frac{(O - E)^2}{E}$$

where w_i is one of the four w values (w_c, w_c', w_l, w_l').

The final chi-square test value for w is calculated using:

$$\chi^2(w) = \sum_{i=1}^{n} \chi^2(w_i)$$

The larger the value of this term, the more significant the difference between the number of occurrences of the word (or word-pair) is in the articles of C and L.

A threshold value (t) is defined to extract only those words (or word-pairs) that are significantly more in one than the other. Words (and word-pairs) in C and L whose χ^2 value is greater than t are withheld and the rest are discarded.

Once this is done for each word (or word-pair), they are sorted in descending order of the χ^2 value. From this sorted list, the first z words (or word-pairs) that occur more in the one kind of articles than the other are identified.

Retrieve sentences containing these words and separate them into 2 columns

Using the z words or word pairs obtained from C, sentences containing these words in the articles in C are searched. The same is done for L. The sentences obtained from C and L are compiled into separate columns of a table and displayed back to the user. Only words that occurred a lot on one side of the table but rarely on the other side are displayed.

At the time of parsing the articles, a repository of all the sentences and the articles to which they belong to is created. For every significant term, this repository is searched to find sentences containing the different forms of the word/word-pair.

A word that occurs frequently may also occur frequently with another frequent word. For example, suppose that the words `stem` and `cell`, as well as the word-pair `stem cell` occur frequently. We don't need the single-words `stem` and `cell` as top words, as the sentences that will be retrieved will pretty much be the same as the ones retrieved for the word-pair `stem cell`. Obviously, a frequent bi-gram is more interesting that a frequent word. To implement this, two things need to be done: a

single-word is displayed only if it is not already being shown as a part of a previous bi-gram; when a bi-gram is displayed, single-words that are already displayed which are part of this bi-gram have to be removed. This has resulted in more non-redundant sentences. However, 'grouping of similar words' makes the problem more challenging. A word group may contain a single word and a word pair as similar words (example: `marrow` and `bone marrow` are grouped into one word group). Therefore, the word group is first checked against all existing significant bigrams to make sure that none of the unigrams are already part of a existing significant word groups (in our example, `marrow` is checked against all previous significant word groups). If it passes this test, the word group is added as significant. Then, the word-pairs in the new significant word group are compared with the previous significant word groups to verify that any existing unigram is a part of this bi-gram (in our case, `bone marrow` is tested to check if a `bone` or a `marrow` already exists). If such a word exists, it is removed from the significant word list, i.e., it is not displayed as a high-rank word even though it has a higher chi-square value than the current word.

There are instances when a word (or word-pair) occurs very frequently in a single article (example: name of a person), thus increasing its chi-square value, although it may not be a true high frequency word. For example, the name of a person may occur several times in just one article; resulting in the word showing up as a top word. This is undesirable. The ideal thing to do would be to somehow take the number of articles that the word (or word-pair) appears in when calculating the chi-square value. A restriction is set up that a word or bigram must appear in at least 3 articles to be considered to be displayed among the significant words.

Once the list with the desired number of significant terms is compiled, the subsequent terms are still read. Only bigrams that encompass existing words are fetched. For example, if the term `rate` is a significant unigram, a term like `crime rate` is preferred even if it is present in fewer articles or has lower chi square value. If such word pairs exist, they shall replace the existing unigrams, because bi-grams are more interesting. These word pairs are displayed even if they are present in only one article. For example, even though `rate` may be present in three articles, and `crime rate` may be present in only one, `crime rate` shall replace `rate`.

For every sentence that is picked for display, a link to the corresponding parsed version of the article is provided. This new page will highlight the query terms, the selected sentence, so that the user can read the relevant paragraphs of the article containing this sentence. Another link to the actual news page in the provider's site is provided if the user is interested in reading the unabridged version of the article.

It is interesting to note that there are words/phrases that occur frequently along with a word/bi-gram. Words/phrases that occur in the same sentence as a significant term, among all sentences containing the significant term on both sides are found. These terms are highlighted using a different color.

Grouping sentences and terms

For every significant word/word-pair, the set of sentences containing the word/word-pair is replaced by a single sentence that represents the entire set. To determine the representative sentence, the similarity of each sentence with respect to all other sentences in the group is calculated using the cosine function. The sentence which has the largest sum of similarities with all other sentences in the set is the sentence that represents the set. A link is provided next to the sentence, which will show/hide the other sentences.

Sometimes, two or more significant terms will contain the same sentence as its representative sentence. In such cases the terms are merged together and the sentence is shown only once. A link is provided, which, when clicked on, will separate/merge these significant terms.

Because of these processes, the significant terms will be displayed out of order with respect to their chi-square values. The results are re-ordered in the descending order of the number of sentences present in the significant side of the table. Therefore, the significant term with the most sentences on the significant side will be displayed first.

Words occurring frequently on both sides

Words with high chi-square values help differentiate the articles since they occur frequently on only one side – either Conservative or Liberal. There will be words that occur frequently on both sides. Such words are also interesting because they talk about the subject matter that is being discussed on both sides.

To achieve this, each word (or bigram) that is read from the articles is taken and its term frequency is calculated from articles on both sides – Liberal as well as Conservative. These words are arranged according to the descending order of their term frequencies. The first few words from this sorted list are taken. From this list, the words that occur in the query string and those words and word-pairs that appear with high chi-square values are removed. Thus, we obtain the list of words that occur frequently on both sides. These words (or bigrams) are highlighted in the results page.

A legend is provided at the top of the page to help users decipher the meanings of the different colored highlighting that appear in the document.

Determining the Sentiment

It is interesting to know if the discussions in the articles on either side of the table are ‘for’ or ‘against’ the subject in the query. A training set of sentences is taken and tagged manually as ‘for’ or ‘against’. From these tagged sentences, a list of ‘for’ and ‘against’ features is created. Using this list as reference, the test sentences are tagged as ‘for’ or ‘against’. To determine the overall sentiment of liberals and conservatives, the following value is calculated for each side:

$$\frac{Number\ of\ 'for'\ sentences}{Total\ number\ of\ sentences\ in\ the\ group}$$

Whichever side has a higher value will be tagged as 'for' and the other side will be tagged as 'against'.

Ideally, this processed should be carried out using a classifier. However, it has been found that this approach tags the two sides of the table accurately in known topics such as 'embryonic stem cell research', 'gun control', 'abortion', etc.

Running the application

Run the `runme.bat` file to start the application. Type the query to search for and the number of sites to fetch for each meta-search engine type (C and L). The output of the program is in a table of a HTML document.

Other works of interested

There is a web site called Skewz (9), which is a place to read conservative and liberal stories selected by users who join a community and vote to reveal the bias in the news. The users can submit new articles or comment on existing articles. The top news articles that are submitted by many users are displayed in the home page. Articles can be searched using an interface provided. Every article is associated with some kind of a meter which displays how skewed it is from being neutral and which way – liberal or conservative side – the article favors. Registered users can comment on existing articles and suggest how skewed it is and which way (liberal or conservative) it sways. The value displayed on the main page is probably the average rating value provided by the article's commenters. Unlike this website where users try to determine the opinion of the articles, we attempt to automatically determine the most significant terms used by liberals and conservatives for any specific query.

Works Cited

1. Metasearch engine. *Wikipedia, the free encyclopedia.* [Online] http://en.wikipedia.org/wiki/Metasearch_engine.

2. MySearchView. *My Search View.* [Online] http://128.226.72.71:8080/MySearchView/index.html.

3. MySearchView: A Customized Metasearch Engine Generator. [Online] June 2007. http://www.cs.binghamton.edu/~meng/pub.d/sigmod028d-meng.pdf.

4. Stanford Log-linear Part-Of-Speech Tagger. *The Stanford Natural Language Processing Group.* [Online] 2006. http://nlp.stanford.edu/software/tagger.shtml.

5. Stemming. *Wikipedia, the free encyclopedia.* [Online] http://en.wikipedia.org/wiki/Stemming.

6. The Porter Stemming Algorithm. *Porter Stemmer.* [Online] 1997. http://tartarus.org/~martin/PorterStemmer/.

7. WordNet. *WordNet, a lexical database for the English language.* [Online] http://wordnet.princeton.edu/.

8. Pearson's chi-square test. *Wikipedia, the free encyclopedia.* [Online] 1983. http://en.wikipedia.org/wiki/Pearson%27s_chi-square_test.

9. *Skewz.* [Online] http://www.skewz.com.