SEARCH REPORT

Report on Social Lookup –
An up to date information retrieval system that makes use of social media sources

Aoife De Buitléar – 12376926 – aoife.debuitlear2@mail.dcu.ie
Emma Duffy – 12495958 – emma.duffy36@mail.dcu.ie
CASE4 – CA4009
11/12/2015

A report submitted to Dublin City University, School of Computing for module *CA4009*: *Search Technologies*, 2015/2016.

I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy.

I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.
By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name(s): Aoife De Buitléar & Emma Duffy

Date: 11/12/2015

Contents

Abstract

The following report describes a social media information retrieval system – “The Social Lookup”. It explains the overall functionality and implementation of the system. The Social Lookup is a social media information retrieval system that uses social networking sites as well as factual news articles and other social media articles to satisfy the users query.

The aim of the Social Lookup is to develop a real-time information retrieval system, lessen users work load with regards to search and fill the need for social media search technologies. It is not required that the user have any background in query based search as the system will be designed with all manner of users in mind.

This report should enable a reader, with a background in search, to understand the high level design of the system to the point that actual implementation is possible.

1. Overview

The information retrieval system in question will return up to date, relevant information from queries inputted by the user. It will retrieve information from social media sites containing news and entertainment articles such as “Her.ie”, “The Onion” and “Football365”. It will also pull relevant tweets from the social networking site Twitter.

This method of search will enhance a users’ experience of information retrieval for current affairs. The key goal is to make life as easy as possible on the user by retaining current and relevant information in the one place.

Our proposed system is unlike any we have researched in social media search, as it not only retrieves information from social networking sites but also factual information from news sites and published articles. Additionally, a unique feature which we have incorporated into our system is that the user has the option to enter a query for any non-specific topic or will be able to refine their search to a specific category or topic of their desire. As a result of this, there will be an increase in the chances of returning relevant results from users' queries. The system will enable users to search for topics that they are interested in and retrieve multiple, relevant, articles and tweets to read from. These documents will have different authors which in turn allows for different views and opinions in order for the user to broaden their own views.

The search system will work as follows:

- User selects from a list of categorised topics e.g. Sport, music, news, ALL or Trending.
    - If Trending is selected, the user can view current topics trending via Twitter.
- User inputs search query
- Query is searched for under specified topic selection criteria.
- If “All” is selected, search query does not search within specified topic, instead it searches all possible criteria
- Retrieved document extracts are then returned for the user to view

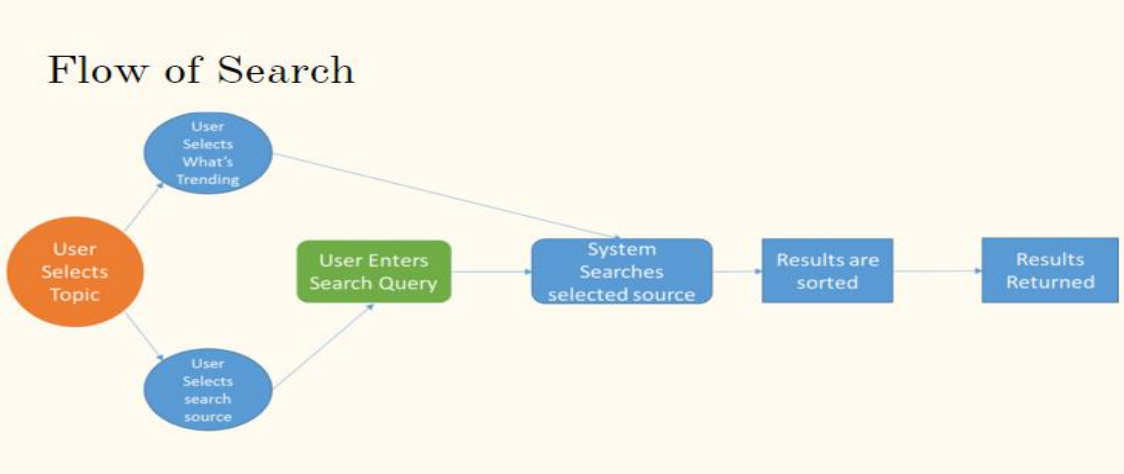


Figure 1 - Flow of search

The results are returned in newest-first order. The newest, most recently posted documents, are assumed most relevant.

2. Functional Description

2.1 Architecture

In order to ensure user satisfaction with regards to relevant retrieval, we have developed an architecture that should maximise relevant results returned by the IR system. We want to take careful consideration when developing our search system in what requirements are needed, who will be using the system, and what data will be pulled?

2.1.1 Requirements

The requirements of the proposed system will depend mainly on what the user will need/want to satisfy their query. Users of search engines typically look for specific features, some of which include:

- Interactive processing of search queries
- Effective ranking of relevant content (particularly at the top of the retrieved list)
- Users are generally not willing to click “next page” button. This motivates the need for high precision at the top end of the ranked list and very rapid response time is essential.

2.1.2 The User

In order to cater for the movement of the user base of Information Retrieval technologies from trained professional information seekers to general users with no training in IR systems, our system will accommodate users of all characteristics and backgrounds. This in turn, will ensure that

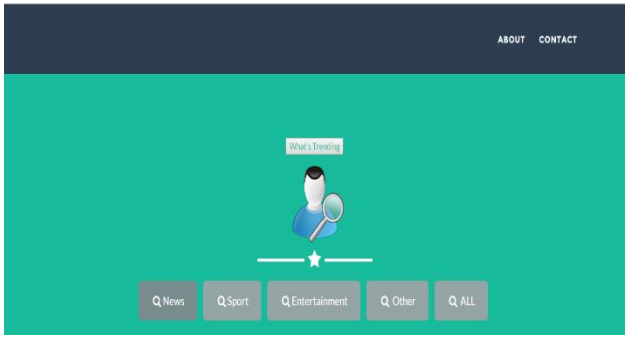


Figure 2 - Sample User Interface

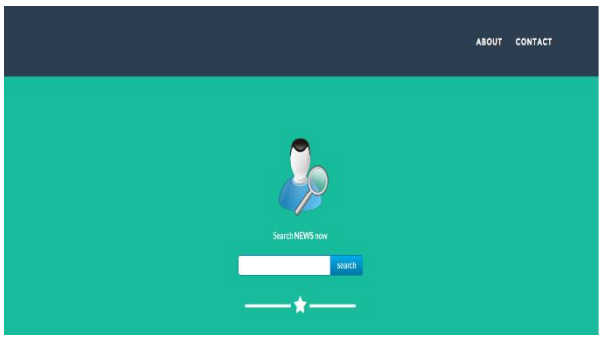


Figure 3 - Sample User Interface search page

no bias is involved in our search system with regards different levels of experience with technology, search and using IR systems or IR technologies. (Croft, 1995)

In the images above, *Figure 2* and *Figure 3,* we have provided an outline of "The Social Lookup". In terms of design, we incorporated into our system our knowledge and experience of HCI techniques to ensure we developed an easy-to-use, user-friendly and accessible UI. We ensured that a user with no previous experience with querying languages would be successful in navigating the UI.

2.1.3 The Data

Due to the fact that our system implements a topic selection feature, the data that will be used in our system will be pulled from numerous different sources as our system does not focus on one specific topic.

Our proposed system entails an aspect of complexity involved with regards to its successful functionality. To ensure that we have effective Web Retrieval, we will need IR methods that give high precision for the very large document collections that we will be working with. Thus, we need measures of document importance which go beyond the matching score between the query and the document contents.

The main objective of our entire system is to recognise relevant documents with very high precision for queries typically ranging from 2-8 words long. To achieve high precision our system will not rely on exact string matching and using ranking algorithms. For a further in depth description of an IR method which was used (see 3.2 Vector Space Model). (1997)

2.1.4 Components of our IR system

Our search engine will match queries against an index that has been created. This index consists of all of the words in each document, plus pointers to their locations within the documents. This is called an inverted file. (see 2.1.5 Functions of an IR system). (Liddy, 2001)

Our system will also include a recommender system. The recommender system will seek to predict items that will be of interest to a user. It will try to identify items which might satisfy a user's information need, but also unveil items that may be of interest to the user that may not have been retrieved by the IR system itself. (Jones D. G., 2015) (see 3.3 The Recommender System)

2.1.5 Functions of an IR system

Our search engine will have three main functions:

1. Crawling

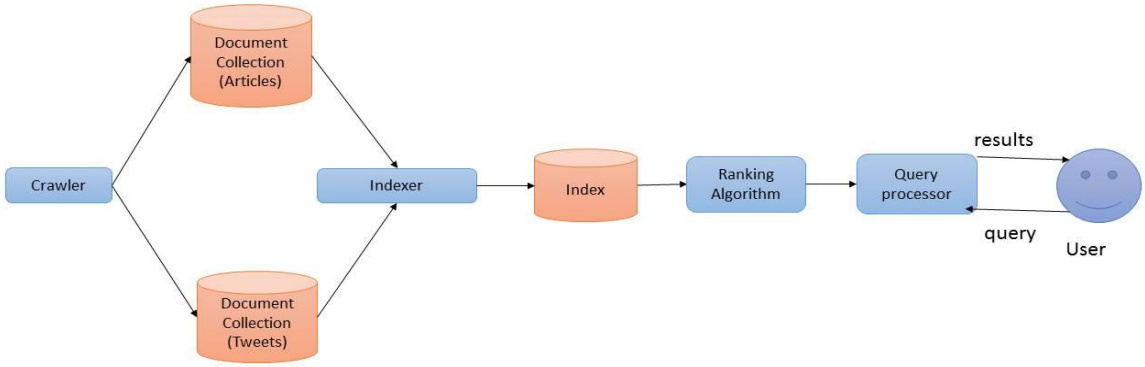
In order to return results and information to our systems' user, information needs to be stored somewhere. In order to achieve this we will need a web crawler. A crawler is a program that visits web sites and reads their pages and other information. Once the crawler have completed the task of finding information on Web pages, the search engine must store the information in a way that makes it useful. This is where our index will come into play. (Anonymous, 2005 - 2015)

2. Building an index

Our search engine will index, collect, parse, and store data. The purpose of storing our index will be to optimize speed and performance in finding relevant documents for a search query. Without an index, our search engine would need to scan every document in the corpus, which would require considerable time and computing power. As our main aim is to satisfy the users' needs, speed performance is a critical issue. (Anonymous, 2005 - 2015)

3. Relevant results

Providing search users with a ranked list of the websites they've determined are the most relevant. (See 3.1 Algorithms)



2.3 Analysis of pros and cons of design:

2.3.1 Pros

- User specified search topics
- Various, wide variety, of documents retrieved
- Multiple authors, therefore multiple opinions and views

2.3.2 Cons

- Twitter and social media articles searched separately – ranking is also separate and unrelated
- Only publically available tweets available for retrieval

2.4 Limitations & Assumptions

The ability to compare the relevance of tweets to articles and in turn to return them in a ranked list, will be a limitation. This is due to the vast difference in document lengths of articles comparted to tweets – tweets have a limit of 140 characters – and so the two types of social media will be searched for and returned separately.

By implementing document length normalisation it may be possible to avoid this obstacle. However, we believe that by keeping tweets and articles separate our system will have a much more appealing User Interface.

An assumption made is that newer documents are more relevant to the users' search. This is because our IR system design is aimed at being as up to date as possible, returning current affairs information.

This assumption may not be true in all cases, which is why we include a document ranking system to return high ranked documents based on the users search query out of most recently published documents. (See Algorithms)

3. Implementation and Evaluation

3.1 Algorithms

Following research into search engine structures and functionalities, we agreed that the following algorithms would be best suited for our search system:

3.1.1 TF-IDF

The system will use TF-IDF (Term Frequency - Inverse Document Frequent) scoring function to determine the importance of each word in a document. Typically, a term that occurs frequently in a document is more important in the document than an infrequent term, this is the TF and the number of occurrences of this term in the document is used as the term weight. However, stop words such as and, the, a, in, etc. can skew results which is why they will be removed to help improve the overall performance and efficiency of the system. (Ho Chung Wu, Robert Wing Pong Luk, 2008)

Long and verbose documents usually use the same terms repeatedly which can also skew the term weight and in turn return an incorrect document rank. To compensate for these effects, *normalisation* of term weights is often use. Term weight normalisation is used to remove the advantage that long documents have in retrieval over short documents. (Document Length Normalisation)

This will be of great importance to our IR system due to the vast differences in document lengths being used, e.g. tweets vs articles.

3.1.2 PageRank & Content-Based Scoring

The system will make use of the PageRank algorithm to rank website pages and measure their importance. PageRank is a family of link analysis algorithms which assign a numerical weighting to each element of a hyperlinked set of documents (such as web pages). It is used by the popular search engine Google, to help determine a page's relevance or importance (Stephan, 2008).

The numerical weight that it assigns to any given element *E* is referred to as the *PageRank* of *E* and denoted by ***PR(E)***
(Anonymous, PageRank, 2015)
Simply put, suppose a small universe of four web pages: A, B, C and D. If all those pages link to A then the PageRank of A would be the sum of the PageRank of pages B, C and D.
PR(A) = PR(B) + PR(C) + PR(D)

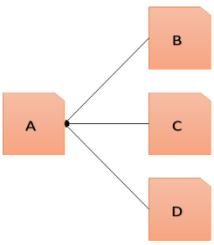


Figure 4 - Basic Page Rank

The system will also make use of content-based scoring – a basic statistical analysis of a document's contents using term weighting.

A document's ranking may improve through a combination of content based scoring and PageRank. The maximum score for a document will be achieved by a document containing all, and only, the words of a search query with high frequency (Jones G. , 2015). The higher the frequency of a search query term occurring in a document, the higher its ranking.

3.2 Vector Space Model

The *vector space model* is one of the oldest and best known of the information retrieval models. In summary, it attempts to determine how similar retrieved documents are to the users query by constructing an N-dimensional token space, where N is the number of tokens from each query. Queries and documents are represented as vectors in a high-dimensional space where each vector

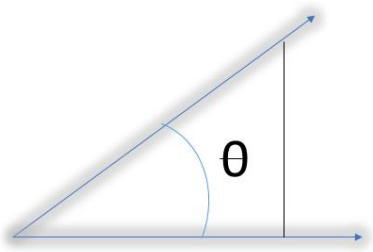


Figure 5 – Vector Space Model

corresponds to a term in the vocabulary of the collection (Stefan Buttcher, 2010). The degree of similarity of a document vector *d(j)* to a query *q* vector is the cosine of the angle between them.

Cosine is a normalised dot product. The higher the term weight, the greater the impact on cosine.

a.b = | a | | b | cos θ

The similarity between the query *q* and each document *d(j)* can be computed as follows and The documents are ranked in decreasing order of similarity.

$$Sim(q, d(j)) = \frac{\sum_{i=0}^{I-1} w_q(i).w_d(i,j)}{\sqrt{\sum_{i=0}^{I-1} w_q(i)^2}\sqrt{\sum_{i=0}^{I-1} w_d(i,j)^2}}$$

Figure 6 - taken from page 84 in Gareth Jones Text Retrieval Notes

These algorithms, combined, will aid in building a functional and well ranked IR system. These particular algorithms are of use to the type of IR system we have designed as we will have a large collection of documents of different sizes and styles to search through.

3.3 The Recommender System

Content-based filtering is a method used to recommend items that may be of interest to a user, based on the user's rating of the items and their contents. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analysing the content of the items which have been seen by the user. (Anonymous, content-based-filtering, Unknown)

With this we aim to develop a recommender system that will recommend relevant articles to users or twitter users that they may wish to follow based on the contents of their tweets.

3.4 Programming Languages

In order to develop the proposed system, we will use numerous languages to succeed. We applied what we learnt from our search labs in the decision of which languages to use. To ensure a user-friendly and accessible UI, as well as functionality, the languages to be used include:

- For the front-end development:
    - HTML, CSS & Bootstrap framework - These will be essential in ensuring user satisfaction with regards easy to navigate layout.

- For the back-end development :
    - Java, JavaScript - for interactivity and communication with websites.
    - Python - for implementing our web spider for web crawling.

4.References

Works Cited

Amit Singhal, Gerard Salton, Mandar Mitra, Chris Buckley. (n.d.). *Document Length Normalisation.* New York.

Anonymous. (2005 - 2015). *What is search engine crawler*. Retrieved from Brick Marketing: http://www.brickmarketing.com/define-search-engine-crawler.htm

Anonymous. (2015, 12 6). *PageRank*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/PageRank

Anonymous. (Unknown). *content-based-filtering.* Retrieved from Recommender Systems: http://recommender-systems.org/content-based-filtering/

Croft, W. B. (1995). *What Do People Want from Information Retrieval?* Amherst: D-Lib Magazine.

Ho Chung Wu, Robert Wing Pong Luk. (2008). *Interpreting TF-IDF Term Weights as Making Relevance Decisions.* New York: ACM Transactions of Information Systems.

Jones, D. G. (2015, November). *Recommender Systems.* Retrieved from loop.dcu.ie: https://loop.dcu.ie/pluginfile.php/784311/mod_resource/content/1/recommender.pdf

Jones, G. (2015, October). *Web Retrieval.* Retrieved from loop.dcu.ie: https://loop.dcu.ie/pluginfile.php/772092/mod_resource/content/2/linkage.pdf

Liddy, E. (2001, May). *How a search engine works*. Retrieved from infotoday: http://www.infotoday.com/searcher/may01/liddy.htm

Slava Frid, Liza Logounova, Alexander Michailov, Oleg Nusinzon, Lenny Zeltser. (1997). *High Precision Information Retrieval with Natural Languuage Processing Techniques.* Pennsylvania.

Stefan Buttcher, C. L. (2010). 02 basic techniques. In C. L. Stefan Buttcher, *Information Retrieval - Implementing and Evaluating Search Engines* (pp. 22-25). Waterloo: MIT Press.

Stephan, D. (2008). *Page Rank Algorithm Explained.* Retrieved from linksandlaw: http://www.linksandlaw.com/technicalbackground-pagerank.htm