MakeMyPage: Social Media Meets Automatic Content Generation

Francisco Iacobelli and **Kristian Hammond** and **Larry Birnbaum**

f-iacobelli@u.northwestern.edu;{hammond,birnbaum}@cs.northwestern.edu

Intelligent Information Laboratory

Northwestern University

2133 Sheridan Rd.

Evanston, IL 60208

Abstract

Finding out about a topic online can be time consuming. It involves visiting multiple news sites, encyclopedia entries, video repositories and other resources while discarding irrelevant information. MakeMyPage aims to speed this process by combining automatic aggregation of information with social media to build web pages with images, videos and links to important information about a topic. MakeMyPage uses automatic aggregation to provide the initial content of the web pages. This content is organized by type: blogs, news, web links, images, video and a main article. MakeMyPage creates a web page by selecting a few items from each category, plus links to more resources within it. Users can vote on the links and media they like best for a given topic and, based on these votes, the system promotes them to and within the main web page. MakeMyPage can be thought of as a collection of wiki pages where people enhance automatically generated content not by editing the text in it, but by voting and suggesting new links. The system's focus is on the organization of content that is genuinely useful and on point. MakeMyPage continuously tracks popular search queries and maintains a database of web pages about these topics.

Motivation

When people want to find facts about specific subjects, their first stop is usually a search engine. Searches such as "what are the symptoms of a common cold" or "who is Fidel Castro" can be typed verbatim into a search engine and the first results are likely to point to correct factual information. However, a parent who is undecided as to whether or not to home-school their children, a college student working on a paper about some topic or a person trying to find information about a potential employer are unlikely to obtain the information they are looking for just by submitting their topic of research to a search engine. People in situations like these are likely to visit many websites and perform many searches before they are satisfied with their research. It is often the case that each website they visit provides only isolated pieces of information about the topics; furthermore, some of the information they receive may be inaccurate or irrelevant. In sum, finding out about a general topic can be time consuming and by no means trivial.

Now, imagine a different scenario where these people type their topic of interest in their search engine of choice and the topmost link in the results points to a web page that is devoted to that topic and displays the most relevant resources (web links, blogs, news, facts, videos and images) about it. Such a web page would have saved these people a lot of time. A broad spectrum of relevant information allows them to decide quickly what subtopics are worth focusing on, which links to explore further and which links to discard.

MakeMyPage is an application that combines social media with automatic aggregation to produce a web page that has both variety of media and good quality content about a topic. Because MakeMyPage produces web pages —not search results— that can be indexed, they will show up in the user's favorite search engine, thus freeing the user of having to look for the same search query in multiple specialized search engines.

MakeMyPage can be thought of as a set of wiki pages where people enhance automatically generated content not by editing the text in it, but by voting and suggesting new links.

In this paper, we provide background related to our aggregation technology and we describe MakeMyPage. We present the results of a pilot user study and we finish by detailing conclusions and future work.

Background and Related Work

Page rank (Brin & Page 1998) and related models are popular methods used by search engines to find relevant links about a topic. However, because these models rely on a network of links, i.e. web pages linking to other web pages, higher ranked web pages require that people take the time to create links to them. This creates a latency effect in which the information that is necessary to rank the pages takes some time to get to the search engine. This latency effect is particularly notorious when the searches are about very recent events. Search engines have been trying to improve ranking algorithms to mitigate this latency effect. They have incorporated factors other than links networks, such as geographic information of their users (Almeida & Almeida 2004), presentation of information in topical clusters (Ferragina & Gulli 2008) or incorporating user feedback, in the form of votes, in the results. Agichtein, Brill and Dumais

(2006) found that adding users' feedback to web searches increased accuracy of the top results by 31%. Recently, search engines such as Google and Yahoo have been implementing user feedback for their results.

Some search engines not only consider user feedback to rank the results, but they also consider user input in the content generation. This is the case of social news aggregators (Lerman 2007a) where users post links while other users vote them up or down. In one study about Digg[1], a very popular social news aggregator, Lerman (Lerman 2007b) mentions that sometimes a topic would be so interesting to users, that activity would spike and, in some cases, news could be posted, voted and ranked before Google News[2] was able to index them.

However, social news aggregators present two important problems for searching content that is on point with regard to a specific topic: The first problem stems from the posting and promotion strategies of those websites. Users post articles with a specific context in mind, however, when postings are voted, they are promoted globally, disassociating them from their original context. Because of this disassociation, search queries can potentially return several links that are not on point with a search. For example, a search on a popular actress's name in Digg and a search on her name plus the word "pictures" may return similar results. That is because people's votes ranked the pictures very high globally, therefore, the pictures are retrieved regardless of the context in which they were posted. Information other than pictures, however, may not make it to the first page of results in either case. Marchionini *et al.*(Marchionini, Capra, & Shah 2008) found this problem with other social media websites as well.

The second problem is the content of webpages and diversity of topics. The content on these sites is driven by a few top users (most active) and those users with larger social networks. Lerman (Lerman 2007b) points out that people are more likely to vote on posts that their friends vote and they also tend to "friend" the top users. It follows from this that people with larger social networks get their favorite sites promoted faster. The converse is also true.

Another technique to improve the information retrieved by search engines is aggregation of media, either automatically (Halevy, Rajaraman, & Ordille 2006; Wright 2008, for example.) or manually (Morris & Horvitz 2007). In automatic aggregation, custom crawlers mine databases and websites refining query terms to extract diverse information. Recently, a few companies have started services that aggregate content in this manner (Bradley 2007). However, these services have a much higher latency effect than popular search engines.

MakeMyPage, in contrast to all the approaches described above, is a system that uses automatic aggregation to generate a web page of resources that are on-point with regards to popular queries, and uses social media to improve the quality of its contents by allowing users to vote those links or suggest new ones. Because the contents are tightly coupled with the generated web page, links are promoted in context.

This results in reliable webpages of popular topics that will tend to stay relevant and show up on the top of any search engine's result set.

MakeMyPage prototype

MakeMyPage web pages are comprised of a main page with relevant links grouped in categories. If any one group has more information than what can be shown in the front page there will be a "show all" link at the bottom of the group. By visiting the "show all" link, users can find additional resources about a topic within that category. People can vote on each piece of information effectively promoting or demoting it in the context of the actual webpage. Figure 1 shows the overall layout of a MakeMyPage.
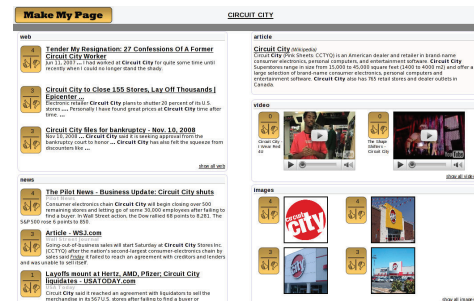

Figure 1: MakeMyPage's main page. Sections are: web links, news, blogs (not displayed), one article about the topic, videos and images.

Initially the content is generated automatically, however, users can suggest links to additional content by clicking a "suggest a link" button. After a user suggests a URL, it is queued for future processing.

MakeMyPage is comprised of several modules that — together— retrieve, process, aggregate and present information. Figure 2 shows its architecture and modules. The modules from the figure are:

Topic Extraction: This is a simple module that collects popular web searches and may reformulate queries that are similar to each other to have a better chance at returning relevant information. Currently, the search terms are taken from Google Hot Trends[3].

Page Display and User Ranking: This module acts as a "traffic cop" among modules, coordinating the resources to build and display web pages. When a new search term is introduced to the system by the Topic Extraction module, or when the user requests a pre-built web page, Page Display asks the database of URLs for the URLs pertaining to that web page. If the term is not in the database of URLs, the Page Display module queries a Content Gathering module that finds and stores information in the

URLs database. The User Ranking module controls the voting by users and the policies that regulate the ranking of links to information.

Content Updating: This module decides whether to build new web pages or update information on existing web pages based on predefined policies. When this module finds a web page that needs updating, it triggers the Content Gathering module to starts the generation of new links for the web page.

Content Gathering and Content Filtering: These are the main modules for content generation. Content Gathering consists of a set of small modules that retrieve specialized information. Each module specializes in retrieving information in one of the following categories: web links, blogs, video, images, news and one encyclopedic article that is relevant to the search query. Content Filtering examines the data retrieved by the specialized strategies and determines which data to retain and which to throw away.

In the next section we describe, in more detail, how the system assembles webpages by gathering content and how the content is improved by filtering and voting algorithms.
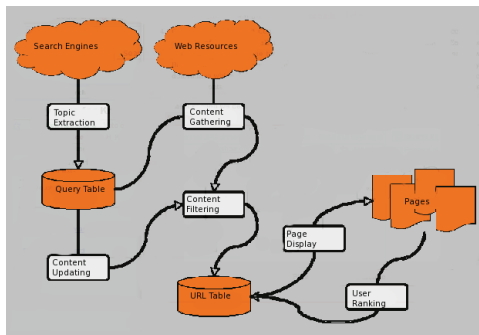

Figure 2: Overall architecture of MakeMyPage

Assembling Webpages

Because popular searches are an indicators of what people consider interesting, We extract popular terms from Google Hot Trends. Our system retrieves the top 100 trends every two hours. Because a topic and the many queries targeted at it tend to stay popular for at least a couple of hours, this period of time maximizes the diversity of topics and minimizes the search terms.

News Articles

News items must be authoritative and complete. It follows then, that news items should be selected from trusted sources. However, often smaller, local news sources are the ones with more information on very recent events within specific communities. Therefore, to retrieve news items, each query is searched against trustworthy sources such as top newspapers and news agencies and smaller news sources from Google news, which tends to returns articles from local news sources. When all the news items come in, MakeMyPage visits each one and extracts the core content section

of the articles. This is done by parsing the HTML tree and selecting the HTML section with most text in it after comments and scripts have been removed and paragraph tags have been consolidated. This method provides, usually, the largest contiguous readable piece of a web page. We also filter the name of the source of the news and disclaimer notices.

After the core content has been found, MakeMyPage displays the first paragraph of the news article. The first paragraph of a news story (also called "lead") tends to be a very good abstract of the news story (Bell 1991).

Blogs

Blogs reflect the opinions of users about a topic. MakeMyPage retrieves blog links, using the Google APIs, and visits those links to extract the section with the most text in them. The algorithm is similar to that of news articles (see the previous section). Because of the pervasiveness of spam blog, MakeMyPage also checks to see if the contents of the blog are written in a more or less narrative way as opposed to advertisement and lists of popular search terms (which is what comprises a large number of spam blog). To determine which texts were written in a narrative way we looked at the ratio of stop-words[4] per word on the texts ($\frac{N_{stop-words}}{N_{tall-words}}$). This decision may seem ad-hoc at first, but we confirmed, by pre-testing on many blogs, that this ratio was a differentiating factor between spam blogs and real content. A list of popular search terms tends to minimize the number of stop-words whereas narrative balances stop-words with content bearing words. Empirically, we found that narrative pieces have a ratio of 0.4 - 0.76 stop-words per word and spam falls outside of this range.

Main Article

Complete information about a topic must, almost always, include encyclopedic information (overview, facts, definition, figures) about a topic or about an entity that is closely related to the topic. This is the function of the main article on a MakeMyPage web page. To retrieve the main article, MakeMyPage performs a series of successive queries that stop when one query returns a result. MakeMyPage starts searching Google for Wikipedia articles using the current search term. If a link is retrieved, then it is visited and the text of the first paragraph is stored. If not, MakeMyPage searches Wikipedia for relevant entities derived from the results obtained so far. MakeMyPage uses an entity detection service [5] on the text that results from the concatenation of the four top web results and the top 2 news articles. MakeMyPage picks the highest ranked entity and searches it using the Wikipedia API.

If there are still no results, the main search term is searched in one last place: The Internet Movie Database (IMDB). Because the initial focus of MakeMyPage is to be good at retrieving popular searches, and many of those involve celebrities or movies, IMDB becomes a good place

---

[4]Stop-words are words that do not bear content, such as very common words, prepositions, determiners and articles

[5]http://www.opencalais.com/

to find this information. If IMDB returns the exact search term as one of its results, MakeMyPage visits the page for that result and stores the first paragraph of the biographical information about it.

Voting and Suggesting New Links

Despite the simplicity of the classification and filtering algorithms of MakeMyPage, the relevance of the links retrieved is usually good. In fact, pilot studies reveal that the content generated is rated equal or better [6] than that of websites with, presumably, more complex algorithms such as Kosmix or Yahoo Glue, and it is rated better [7] than websites where users generate their own content such as Digg or Reddit. However, the quality of the information can be improved in many cases and the results displayed on the main page can also be tailored better to the topic at hand. To achieve this, MakeMyPage allows users to vote on the content of the webpages. When a user votes on a content item, it is promoted or demoted only in the context of the webpage in which it was voted. The number of votes is then leveraged to decide what to display on the main page.

Sometimes direct voting and suggestions are not the only factors that determine what content will be displayed on the main page. In particular, MakeMyPage may rank more recent news with few votes higher than older news with many votes. News ranking is computed by the following formula: $\frac{1}{recency}(posVotes - negVotes)$. Here, $recency$ is the number of seconds that have passed from the publication date of the news article; $posVotes$ is the number of positive votes for the given article and $negVotes$ is the number of negative votes for that article. This method is a simplification that was inspired by Lerman's reverse engineering on Digg's voting policies(Lerman 2007a). The pages displayed by the "show all" links are ranked by votes alone and in case of a tie in number of votes, the most recent link is ranked higher. Videos, images and web links are displayed ordered by the number of votes and and by the date they were retrieved. In this way, videos, images and web links that people like best will remain on the main page. The algorithm relies on a threshold of votes to promote links from the "show all" pages to the main page.

Conclusions and Future work.

MakeMyPage aims to be a new kind of social media system that automatically generates new content collections based on popular demand and then opens the editing process up to the collective judgment of users. Based on the queries submitted to search engines, the system compiles collections of relevant media and makes them available to end users. To do this it makes use of algorithms to ensure that the initial content it retrieves is relevant and it encourages end users to provide content and feedback on the results in the style of Digg and other social media sites.

Future work on MakeMyPage involves disambiguating content within pages, disambiguating popular search queries and continue work on updating policies, voting schemes and algorithms to improve the relevance of the contents of the webpages.

We hypothesize that because the content will on-point with regard to popular queries, the pages produced by Make-MyPage will tend to percolate to the top of the result sets. Because they contain valuable content, they will tend to stay there. This is a departure from the model of aggregators that double as search engines.

References

Agichtein, E.; Brill, E.; and Dumais, S. 2006. Improving web search ranking by incorporating user behavior information. In *proceedings of ACM SIGIR '06*, 19–26.

Almeida, R. B., and Almeida, V. A. F. 2004. A community-aware search engine. In *Proceedings of the 13th international conference on World Wide Web*, 413–421.

Bell, A. 1991. *The Language of News Media*. Language in Society. Wiley-Blackwell.

Bradley, P. 2007. Search Engines: New Search Engines in 2006. *Ariadne, ISSN 1361-3200*.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* 30(1-7):107–117.

Ferragina, P., and Gulli, A. 2008. A personalized search engine based on web-snippet hierarchical clustering. *Softw. Pract. Exper.* 38(2):189–225.

Halevy, A.; Rajaraman, A.; and Ordille, J. 2006. Data integration: the teenage years. In *Proceedings of the 32nd international conference on Very large data bases*, 9–16.

Lerman, K. 2007a. Dynamics of collaborative document rating systems. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, 46–55.

Lerman, K. 2007b. User participation in social media: Digg study. In *International Conference on Web Intelligence and Intelligent Agent Technology*, volume 0, 255–258.

Marchionini, G.; Capra, R.; and Shah, C. 2008. Focus on results: Personal and group information seeking over time. In *HCIR2008*. Microsoft Research.

Morris, M. R., and Horvitz, E. 2007. Searchtogether: An interface for collaborative web search. In *Proceedings of ACM UIST*, 3 – 12.

Wright, A. 2008. Searching the deep web. *Communnications of the ACM* 51(10):14–15.

---

[6] Not statistically significantly

[7] Statistically significant at the 0.05 level