Metabrain: Web Information Extraction and Visualization

João Teixeira
Dept. of Computer Science Engineering, IST, TULisbon
Av. Rovisco Pais, 1000 Lisbon, Portugal
joao.teixeira@ist.utl.pt

Gabriel Barata
Dept. of Computer Science Engineering, IST, TULisbon
Av. Rovisco Pais, 1000 Lisbon, Portugal
gabriel.barata@ist.utl.pt

Daniel Gonçalves
Dept. of Computer Science Engineering, IST, TULisbon
Av. Rovisco Pais, 1000 Lisbon, Portugal
daniel.goncalves@inesc-id.pt

ABSTRACT
Nowadays, a hitherto unseen amount of information can be found in the World Wide Web. While available, this information is fragmented among different web sites. This is especially true for implicit knowledge, not directly written in any one site, but arising from patterns and interactions between pages. For instance, the number of search results for a particular query string might be a meaningful indicator of its popularity or overall interest. Our research focuses on the design of an interface that allows end-users to access implicit information. A prototype application, Metabrain, embodies our solutions and makes it possible to mine the web for statistically relevant patterns, with the help of simple and straightforward algorithms and user interface. To help the users make sense of that information, Metabrain then allows custom visualizations to be crafted. User studies show that users can search for relevant information up to four times faster than using traditional Web search engines alone. A system usability scale questionnaire confirms the interface is usable and effective.

Author Keywords
Implicit knowledge in the web, Information Visualization, Human-Computer Interaction, Information Extraction.

ACM Classification Keywords
H.5.2 User Interfaces - Graphical user interfaces (GUI), H.5.m Miscellaneous.

General Terms
Design, Experimentation.

1. INTRODUCTION
The advent of the World Wide Web led to the rise of an information society that increasingly pervades a large part of the world. Especially interesting is the fact that information consumers now also are its creators. No longer do we live in a world where content is produced by a few for the consumption of many. It can be claimed that the WWW implicitly represents our societies, interests, concerns and knowledge. We present Metabrain, a system that uses simple statistical-based techniques to tap that "collective unconscious" and let non tech-savvy users collect and visualize implicit information available online.

AVI '12, May 21-25, 2012, Capri Island, Italy

That implicit knowledge is difficult to ascertain. Resulting from patterns arising from different sources, in no single place can we find it distilled in an easily understandable way. Traditionally, Information Extraction (IE) techniques have been used to garner digests of non-explicitly structured information. ReForm [5] and Marmite [7] provide interfaces for information extraction based on web-site mashups. These and similar works, however, focus on the extraction of individual information items (tables of upcoming events, people of interest in an organization, etc.) instead of trying to infer more generic patterns. Also the IE techniques involved tend to be brittle, very sensitive to changes in the data sources.

Instead of looking for individual factoids, our goal is to leverage on the large amount of available information to find more general patterns. Simply looking at the number of search results found for a specific query can be a good indicator of a topic's popularity. Even more insight can be gleaned comparing the numbers of results for different queries. A Google search for "*Cristiano Ronaldo football*" (about 73,900,000 results at the time of the writing of this paper) returns a much larger number of results than a search for "*Cristiano Ronaldo dance*" (about 19,100,000 results). It is easy to conclude that Cristiano Ronaldo is more connected to football (he is a famous player) than to dancing. It is not the absolute numbers that matter, but their relative values: one is clearly more frequent than the other.

Several works try to make such implicit patterns apparent. "What's in Wikipedia" [3] uses the online encyclopedia's own article categorization scheme to determine the most important and active subjects. Prism[1] establishes the colors more strongly associated to a concept by pairing the concept with color names in query strings and looking at the results. The use of simple, statistical and heuristic methods grants them a greater robustness and scope than traditional custom-tailored solutions. However, they still possess one critical flaw: while their scope is wide, they require programming skills to be adapted to other situations. Metabrain provides a simple yet flexible and effective system to combine data gleaned from different methods, in different domains, without requiring programming skills by users.

Another aspect that we needed to take into consideration was how to help the users to make sense of the data. Using Information Visualization techniques will make this a more amenable task. Two representative solutions that try to provide ways for non-programmers to devise effective visualizations are ManyEyes [6] and Tableau Public[2]. One is very easy to use but rigid, the other is highly customizable but complex. Our efforts aim to get the best of both worlds, an easy and customizable way for users to visualize our extracted information.

[1] http://nodebox.net/code/index.php/Prism
[2] http://www.tableausoftware.com/public

2. THE COLLECTIVE UNCONSCIOUS

We would like to be able to extract information by using generic, statistical methods, avoiding natural language processing and similar domain-sensitive approaches. The goal is to have robust generic methods, applicable regardless of the domain, and don't require fine-tuning or adaptations if something in a web site's structure changes (something that is likely to happen sooner or later, given the dynamic nature of the world-wide-web). We will make use of search engines in most of our methods since they have gone half-way, by crawling and indexing "the entire" world wide web. We can focus on using those indexes to our advantage, making explicit what before was implicit and hidden from sight. We use three different kinds of techniques, described below.

The first method bases itself on ***the numbers of results*** returned by search engines for a given query. It makes more sense for queries about specific concepts, events or people, but can be extended to other things, in particular domains. This result can be used as a measure of *Popularity* (a search for "cats" yields 415,000,000 results and one for dogs gives us 610,000,000 showing that dogs are more popular than cats); as *Measure of Validity* (a query for "I sent it to him" yields 15,200,000 results and for "I sent it to he" only 398,000, so first is probably correct); and to establish *Relationships Between Concepts* (if we want to know what color (for instance, red or yellow) is more often connoted with bananas, we can look for "yellow banana" and "red banana" and compare normalized result numbers.

The second method is based on ***Lexico-Syntactic Patterns.*** By searching for incomplete phrases following pre-determined patterns, we can use search engines to *extract lists of relevant completions*. We have implemented a variant of the algorithms described in [2] and [1]. Imagine we search for `"colors such as *"`. We get results containing expressions like `"(…) if you use colors such as red, orange, and green (…)"`. We can then use automatically inferred regular expressions to find those patterns and, in particular, the words that complete the pattern.

The third and final technique is based on ***Term Co-Occurrence.*** With the rise of micro-blogging usage, it is now possible to more easily extract the general Internet "feeling" on a given concept by looking at what words co-occur with that concept. With this in mind, we implemented a *Sentiment Extraction* module, based on [4]. It then looks for the search terms in the results, and the co-occurrence of words in a list of terms classified according to their underlying sentiment (positive; negative; neutral). We use the Subjective Lexicon[3]. For instance, if we find the word "flower" together with others like "love" and "like", we'd be led to conclude that, generally speaking, people like flowers. To account for situations where the text actually reads "I do not like flowers", we look for denial adverbs and invert the value of the co-occurring words if one is present.

The methods mentioned above all give us, on average, good results. They are purposefully simple, often resorting to heuristics, to render them domain-independent and more robust. As such, they are not guaranteed to provide 100% accurate results. Some methods will work better on some cases than others. The users must exercise their critical judgments when analyzing the results. Also, it must be noted that looking at values such as the number of search results can be misleading. Frequency and popularity aren't necessarily equivalent, nor are co-occurrence and correlation. Also, there is no guarantee that the results lists are complete. Our goal is not to provide an accurate information extraction tool, but rather, to allow the exploration of implicit knowledge by non-programmers. It might not even make sense to talk about a "complete list of results" when dealing with informal, non-structured domains.

[3] Available at http://www.cs.pitt.edu/mpqa/

3. THE METABRAIN PROTOTYPE

Metabrain was created in Python with an HTML5 front-end. We have implemented all the methods described in the previous section. A plugin-based architecture allows us to easily add new ones, if necessary. All plugins access the web indirectly, through a module that communicates with several services using their APIs and deals with limits to the number of requests that can be made to those services. It implements a cache so that results from queries that have been made before can be reused. Currently we can get data from Google, Yahoo, Twitter, Facebook, Panoramio, Wikipedia, Google News, and Google Images.

We tried to simplify every possible step of the information collection process. By default all customization options (source, number of results to consider, etc.) are hidden, preset to a sensible default value. In a more straightforward usage, all that the users need to do is to select what they want to extract. They can choose between different "Input Types": "Extract Geo Location"; "Related Words"; "Suggest"; "Extract By Domain"; "Number of Results"; "Social Trend"; and a general-purpose "Extract". Those types correspond to the extraction methods described in the previous section, albeit with more user friendly names. Each has a short description and examples of possible input values, to help novice users understand them. Figure 1 shows the extraction of zodiac signs using the "Extract by Domain" module.

Results are grouped in a table, with columns that depend on the type of information that was obtained. Simple instance lists give us the instances and their frequency; for locations, we get an additional column for their geographic coordinates; etc. The user can manipulate this table filtering the results, if necessary.

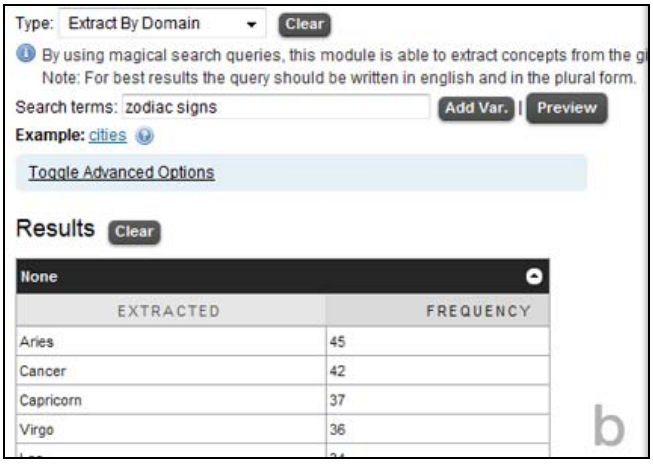

Figure 1: extracting a list of zodiac signs.

Multi-level extraction is possible, feeding the results of one query into another. Imagine a user wants to know the popularity of different cities. He could create a query to extract instances of cities, and then feed those results directly into the popularity module. The resulting table would contain a list of cities and their popularity. If, at any time, the user chooses to use the results of one query as the basis for another, the interface will dynamically

add a new input section. There, the second search query can be defined, based on the results of the first (that can be used as variables in the query string of the second). The variables are textually represented using the percentage sign ("%1", "%2", etc). Graphically, subordinate queries appear indented below the one in which they are used. Complex queries can take several minutes to complete. Thus it is possible to perform limited runs and inspect a preview of the results, before running them in their entirety. The resulting data can then be visualized using several different techniques. The interface for this was entirely developed in HTML5 with the help of the Protovis visualization toolkit. The resulting visualizations are dynamic, support tooltips, and can be exported and embedded on web pages.

A number of different visualization techniques are available, ranging from bar charts to treemaps, scatter plots and geographical maps (Figure 2), among others. Thumbnails facilitate their recognition and provide examples of what to expect. The visualization list is populated based on the types of the columns for the dataset being visualized, making it impossible for the user to select non-applicable techniques. The visualization type can be changed at any time. To the extent in which this is possible, the configurations (what data to display where, colors, etc.) are kept between visualizations, allowing the user to try out different alternatives without losing context.

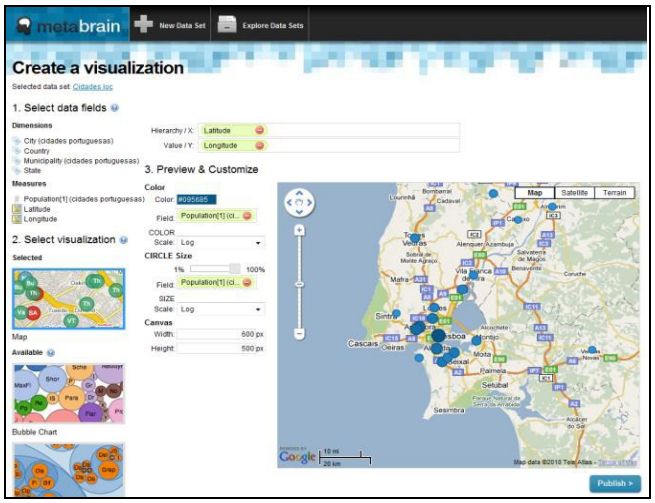

Figure 2. Creation of a map visualization.

The user can choose which columns to visualize by using drag and drop. On the left side of the application a vertical list of column names is visible. They are divided by the type of data they contain: dimensions (independent data, such as nominal, temporal or geographic data), and measures (dependent data, such as ordinal and continuous variables). On the top right, all degrees of freedom allowed by the currently selected visualization technique are displayed (X-axis, Y-axis, height, bubble size, area size, color, line width, etc.). By dragging a column name to a field corresponding to a degree of freedom, the values in the corresponding column will be bound to it. This is highly customizable, being possible to choose color palettes, linear or logarithmic scales, and so on.

4. EVALUATION

We carried out a series of user tests whose purpose was to estimate if users were able to use Metabrain efficiently to extract and explore interesting information. This experimental evaluation was carried out by 12 test subjects, (9 male, 3 female), with ages between 19 and 30. All subjects were proficient with web browsing, search engines and commonly used tools. Network quality at the onset of each test showed little variation, with the available bandwidth never falling below 10Mb/s. After a 15 minute introduction and demo, in the next 40 minutes the user was asked to do two sets of tasks, one related to information collection and another to information visualization. Each task was performed twice, one using Metabrain and another using an alternative solution: unrestricted internet search for data collection, and Tableau Public for visualization. The users were given a short tutorial and demo on this tool, similar to the one for Metabrain. The order in which each user performed the tasks varied, ensuring half performed the tasks using Metabrain first, and the other half used the alternative methods. Tasks that took longer than 3 minutes were classified as failed and are not taken into account in the following result analysis.

4.1 Information Collection Tasks

These tasks are related to the retrieval of information from the Web. A set of five tasks allowed us to evaluate the performance and usability of our solution in all key facets, especially the ability to execute multi-level extractions. Table 1 shows the values we found. Unless noted, all results were confirmed to be statistically significant with t-tests (95% significance).

Task 1.1 asked the user to find a list of five low carb foods, and another of five high carb foods. We aimed at finding if the users can perform the simplest extraction tasks successfully. Indeed, all users completed these tasks. The task was much slower to perform using web search than with Metabrain, even if it was the users' first contact with the system. On Metabrain the task was, on average, completed almost 4 times faster.

Tasks 1.2 through 1.4 aimed at finding if the users could understand and employ different information extraction methods. In Task 1.2 they were asked to discover what mobile operating system is more popular. The list of systems was given to them at the onset of the task, and all they had to do was to extract the OSs' popularity. Only one user was unable to complete the task using web search. All Metabrain users completed it. Using Metabrain was over 3 times faster than using the web. In Task 1.3 we asked the users to find what were the most popular subjects, from those currently shared on twitter, that were related to "love" (at that moment, "Mother's day" and "Justin Bieber"). Outside Metabrain, most users started by opening twitter, searching for "love", and manually browsing the results. All Metabrain users finished the task (one user for web search did not), 1.3 times faster, on average, than the other users. Task 1.4 consisted on finding colors relating to the word "apple". This was tedious and time-consuming, leading to a low completion rate of 50% outside Metabrain. All Metabrain users finished it, 2.6 times faster.

Finally, **Task 1.5** was designed to see if users were able to perform multi-level searches on Metabrain. We asked them to find the popularity of different cities in Portugal. Only 67% of users performing direct web navigation completed the task. As for Metabrain, 83% did so. Furthermore, using Metabrain was slightly slower (although this difference was not statistically significant). This shows that complex queries are, indeed, harder to perform. Still, more Metabrain users completed the task, and needed half the number of steps, an indication that Metabrain can still help the users to perform this kind of task.

	Task 1.1		Task 1.2		Task 1.3		Task 1.4		Task 1.5	
	avg	stdev	avg	stdev	avg	stdev	avg	stdev	avg	stdev
Time Metabrain	15.83	5.64	8.83	6.91	78.33	43.17	28.50	12.26	102.00	47.24
Time Alternative	65.33	28.50	26.83	11.07	106.20	38.96	73.83	43.51	94.75	52.92
#Steps Metabrain	1.17	0.41	1.00	0.00	1.83	1.17	1.00	0.00	2.20	0.45
#Steps Alternative	4.00	1.26	2.50	0.55	3.60	1.52	1.00	0.00	5.50	0.58

Table 1. Information extraction test results.

	Task 2.1		Task 2.2		Task 2.3		Task 2.4		Task 2.5	
	avg	stdev	avg	stdev	avg	stdev	avg	stdev	avg	stdev
Time Metabrain	33.67	25.71	7.67	4.37	62.20	35.07	11.83	15.03	24.83	8.59
Time Alternative	55.50	47.57	29.75	53.51	36.33	10.02	37.17	23.04	-	-
# Steps Metabrain	3.83	1.17	1.17	0.41	5.20	2.77	1.50	1.22	2.67	0.82
# Steps Alternative	4.50	2.59	2.25	2.50	5.00	2.00	5.33	2.16	-	-

Table 2. Information Visualization test results

4.2 Information Visualization Tasks

We chose another group of five tasks crafted to allow us to determine if the steps necessary to create and personalize the visualizations from datasets were easy to understand and use. On Table 2 we can see the results from our tests.

In **Tasks 2.1 and 2.2** asked the user to visualize the same dataset in a bar chart and line chart. Although all users completed Task 2.1 in both tools, the same did not happen for Task 2.2. There, 33% of Tableau Public users were unable to do so.

Task 2.3 asked the user to visualize a set of geographic coordinates in a world map. In our solution, both the Latitude and Longitude, are represented by the same field, which was the only one that needed to be selected to conclude the task, users did not seem to understand this and proceeded to select other fields such as city name and country name. This led to a series of trial and error steps which led to longer task times when compared to Tableau Public. On the other hand users were more prone to conclude the task using Metabrain than the alternative, due to some extra steps needed while selecting the coordinate's fields.

In **Tasks 2.4 and 2.5** we evaluated the customization of the visualization. Our geographical information has two degrees of freedom, but they do not correspond to latitude and longitude. Rather, geographic position is atomic, in our solution, and will occupy just one degree of freedom. We use the other to define the size of each point in the map, instead of using a side option like in Tableau Public. This turned out to be very intuitive and Task 2.4 was performed to an average of 3 times faster on Metabrain. Now the user was supposed to color each point in relation to an extra data field (Task 2.5). The need to access deep level menus on Tableau Public proved to be so complex that no user was able to conclude the task in this alternative. We cannot draw conclusions on relative times. But since all the test subjects were able to do this on Metabrain, we can say that our solution is likely better.

4.3 System Usability Scale Questionnaire

Each subject was asked to fill out a standard SUS questionnaire yielding a score from 0 to 100, reflecting an interactive system's usability (100 being the best possible value). Our solution has an average score of 77.42, which is considered a positive outcome.

5. CONCLUSIONS

Overall results were very positive. Metabrain is on average 1.55 (Collection) and 1.28 times (Visualization) faster than other solutions. Also, novice users were able to complete the tasks more often with Metabrain than without it. This shows that they were able to understand and use it without major hurdles. Overall, the extraction methods we implemented made sense for users, and yielded good results. Clearly, one aspect needs to be improved: multi-level queries. Overall, user tests have shown that Metabrain fulfills our main objective of providing non-expert users with the bility to tap implicit knowledge in the web without the need to write a single line of code. In future versions of the system, we will take another look at the interface for creating multi-level queries, to improve its usability. Also, we will add the notion of time to the system, to allow visualizations on the evolution of some subject.

6. ACKNOWLEDGEMENTS

This work was partly supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds, and by FCT through the PAELife project AAL/0014/2009.

7. REFERENCES

[1] Banko, M., Cafarella, M, Soderland, S., Broadhead, M., and Etzioni,O., "Open information extraction from the web," Com. ACM 51, 12 (Dec. 2008), 68-74.

[2] Etzioni, O. et al., "Web-scale information extraction in knowitall: (preliminary results)", In Proc. of the 13th int. conference on World Wide Web (WWW '04). ACM, New York, NY, USA, 100-110. 2004.

[3] Kittur, A., Chi, E., and Bongwon Suh, "What's in Wikipedia?: mapping topics and conflict using socially annotated category structure," in Proc. CHI'09, pp. 1509-1512. 2009.

[4] Kramer, A., "An unobtrusive behavioral model of gross national happiness". In Proc. CHI '10. ACM, New York, NY, USA, 287-290. 2010.

[5] Toomim, M. et al. "Attaching UI enhancements to websites with end users." In Proc. CHI '09. ACM, New York, NY, USA, pp 1859-1868. 2009.

[6] Viegas, F., Wattenberg, M., Van Ham, F., Kriss, J., and McKeon, M., "Manyeyes: a site for visualization at internet scale," IEEE Transactions on Visualization and Computer Graphics, pp. 1121-1128, 2007.

[7] Wong, J. and Hong, J. "Making mashups with marmite: towards end-user programming for the web." In Proc. CHI '07. ACM, New York, NY, USA, 1435-1444. 2007.