Poirot: a relevance-based web search agent

José M. Ramírez
Jordi Donadeu
Francisco J. Neves

Grupo de Inteligencia Artificial
Universidad Simón Bolívar
http://www.ai.usb.ve
jramire@ldc.usb.ve

Abstract

This paper describes the first implementation of POIROT, a web search agent based on relevance, that determines the users interests inspecting the pages bookmarked in the web browser and extracting keywords using some information theory methods such as TF-IDF. The keywords are used to build a training set that is processed by an Inductive Logic Programming (ILP) algorithm that learns what is "relevant" to the user. The rules generated with ILP are used to expand user queries and to rank the results. POIROT also models the behavior of the more important Internet search engines to determine which one to use depending on the topic to search. One important design consideration of POIROT is to build its models without asking the user for feedback, from this perspective POIROT is an active learner. Some comparisons with Metacrawler are reported, showing that POIROT outperforms in terms of relevance and precision of the results presented.

Identifying users interests

The approach followed by most of the web agents that try to model users is to present an initial questionnaire and ask for feedback from the user after each action or decision (Chen and Sycara 1997, Newell 1997). This approach tend to be molest and extremely poor to gather hidden relations between features or progressive shifts in user interests. One of our main motivations was to identify user interests, and model the users and search engines, without issuing questionnaires or asking the user for feedback.

The monitoring of users actions seems to be a promising approach, but there are several issues that difficult its implementation. First, the capture of events in a web browser, as save, print or mouse actions, is not an easy task; and second, its difficult to assign the correct meaning to users actions during web browsing. For instance, its not clear how important is to follow a link or to "view" a page for a given time period. A page may stay in the browser's window unattended while the users are taking a break, and that doesn't mean that the page is of interest for the user. The frequency and numbers of visits to a page is a factor that indicates the interest in that page and, more important, interest on the topics related with that particular page (Tauscher and Greenberg 1999).

One option to capture the features mentioned was to develop a web browser or a plug-in that monitor the user activity and reports it to a feature extraction module. The problem is that the plug-in "lives" during the execution of the web and it is limited to act in a single window at a time, limiting its monitoring possibilities. To develop a web browser, given the installed base of the most popular browsers (Sullivan 199b), doesn't make sense.

Another option considered was to use the cache files stored in the proxy server or local hard disks. The cache maintains a log of visited pages and stores the components that form the pages. But pages in the cache are not necessarily pages of interest, they just reflect the navigation pattern of the user.

Popular web browsers give the user the possibility to bookmark pages for future reference. Some web browsers give also the option to organize the bookmarks by topic. Given that the bookmark is an user action that reflects the interest in a given page, we can assert that, at a given time, the pages bookmarked were relevant for the user in terms of general utility, quality, personal interest, frequency of use and potential of future use (Abrams and Baecker 1999).

According to (Abrams 1997), 84% of the users use bookmarks, hotlists or favorites lists as a reference to pages that they liked. The bookmarks are tools of natural use for the user that face common problems of Internet as information overload, disorder, redundancy, lack of quality (Tauscher and Greenberg 1999, Lawrence and Giles 1998 and 1999, Baeza 1998). The bookmarks help to reduce the entropy inherent to Internet.

Modeling users and search engines

Given the decision to use bookmarks as the primary source of information about user interests, the next decision was to develop an agent to extract the information, build a model and interact with the user.

The first problem in the design of the agent was to identify the features that determine that a given page is of interest for the user. The interest of a user, as we mentioned above, can be determined by several factors, but all of them are associated with the topics covered by the page, so we need to determine what are the topics covered by a given page in order to determine if that page is relevant for the user and its degree of relevance.

The topics of a document are not necessarily written explicitly, for instance, a page of a college may mention words as "university", "faculty", "schedules" but the term "education" may not appear. Moreover, if that term appears, how to determine that is related to the topics of the page?. Clearly, the topics covered by a web page are determined from the words that are written in the page, the problem is to rank the words in the page according to the relation with the topic of the page.

One alternative is to infer the topics covered by a document given the frequency of appearance of certain words in the document and a relation established between those words and the topics. In other words, the words that are more related with the topic of a document should be more frequent than others, and pages that cover same topics should share the same set of frequently used words.

(Joachims 1998) suggests to use a feature vector consisting in a word counting vector. Each word has an associated importance, computed based on the frequency of appearance, that serve as input to a Support Vector Machine (SVM) (Haykin 1999, Burges 1998, Osuna *et al*. 1997, Nilsson 1999, Pontil and Verri 1997) that determines the proper classification of the document. The problem with this approach, in addition to the computational cost of SVM, is the lack of important features as incremental learning that would give the agent the ability to adapt to user changes.

(Salton and McGill 1983) present a method to compute a weighted word counting, using weights determined by an algorithm called TF-IDF (Term Frequency – Inverse Document Frequency) that became a standard in information retrieval.

The general idea is to represent each document as a vector in a feature space, where documents with similar topics should be "closer" in the space that unrelated ones, given a distance metric. Each dimension of the feature space represents a word and its weight. The weights are computed based on the frequency of the word, *TF(p,d)*, and frequency of documents, *DF(p)*. From *DF* the inverse, *IDF(p)*, is computed as shown in (1).

$$IDF(w) = \log \frac{|D|}{DF(w)} \qquad (1)$$

where $|D|$ is the total number of evaluated documents. The weight of a given word in a document is computed using equation (2). The higher the value of *d(i)*, higher is the significance of the word within the document.. Note that the significance of a word within a document decreases if *DF* increases, given that if a word is very frequent in a set of documents, that word won't help us to "separate" the documents in the feature space.

$$d(i) = TF(p_i, d)\, IDF(w_i) \qquad (2)$$

As an adaptation of TF-IDF to our particular problem, we assign additional weights to the words according to their position in the HTML document, higher for words appearing in the "keyword tag", titles, headers, etc. and lower for normal text.

Once the words in each document are ranked, we need to infer rules that, starting from these initial documents (the pages bookmarked by the user), help us to determine the relevance of new pages. This problem can be addressed as inductive concept learning, where the concept to learn (right-hand side of the rules) is "relevance" and the conditions (left-hand side) are logic expressions based on topics found in documents. The pages bookmarked by the users are used as the training set, they are all positive instances of the concept.

Inductive Logic Programming (Bergadano and Gunetti 1995) was selected, instead of other inductive methods as Neural Networks, to infer rules due to its ability to derive rules of variable number of conditions, resulting in specific rules (many conditions) that give us precision and general rules (few conditions) that help us to maintain coverage in the search. We limit the number of topics to appear in the left-hand side of rules to 10, to avoid the over-fitting of the agent. The topics with higher ranking are used to build rules of the form:

If ($topic_1$, $topic_2$, … $topic_n$) **then** Relevant **with** RF

Where **RF** is the relevant factor computed from the ranking of the topics included in the rule.

Once the initial rules are generated, ILP techniques are applied to that rules to generate aggregated rules that are also incorporated to the rule base. The result is a rule base with rules extracted directly from documents and more general rules that apply to the entire set. This maintains the required balance between precision and coverage that is shown in the results.

The last issue to address is the modeling of the repositories, search engines in our case; allowing the agent to select the search engine with higher probability to give good results for a particular topic. We use a strategy similar to TF-IDF, maintaining a vector for each

topic, representing the frequency of appearing in each search engine.

Altavista, Yahoo!, Infoseek, Lycos and WebCrawler where used as the search engines of our agent, given that there is consensus about their high coverage and quality (Sonnenreich and Macinta 1998, Sullivan 1999a and 1999b, Hock 1999, Search Engines: Education, information and great links 1999). We need to use several search engines, given that the overlap of pages indexed by the search engines is low (Sullivan 1999). The problem with this decision is that the agent must understand the protocol used by each search engine to receive queries and integrate the results in an unified interface.

Architecture of the agent

Figure 1 shows the architecture of POIROT.

The Topic extraction module reads the bookmarks file and visit each URL to extract the topics of interest for the user. This module reads the HTML of all the bookmarked pages, discard HTML code and stopwords (Cambridge Scientific Abstracts Stopword List, Library of Congress 1999), and compute the count of words to be stored in the table "URL/Topic"

The search engines module receives the expanded query and the selected search engines, according to the topics in the query, and sends the requests to each search engine using different threads. Also receive the results from the search engines.

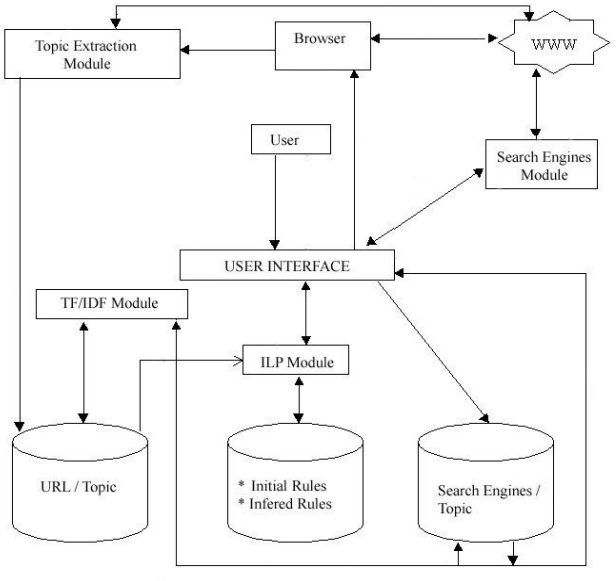

Figure 1 - Agent architecture

The user interface is shown in figure 2. Some important components are:

- Checkboxes to select the browsers to use. This selection supersedes the selection made by POIROT based on the topics of the search. (3)
- Control of how many hits to return. (4)
- Search buttons. (5)
- Keyword edit field. (6)
- Results window. (9)
- Description window. When an URL is selected in the result window, its description is displayed here. (11)

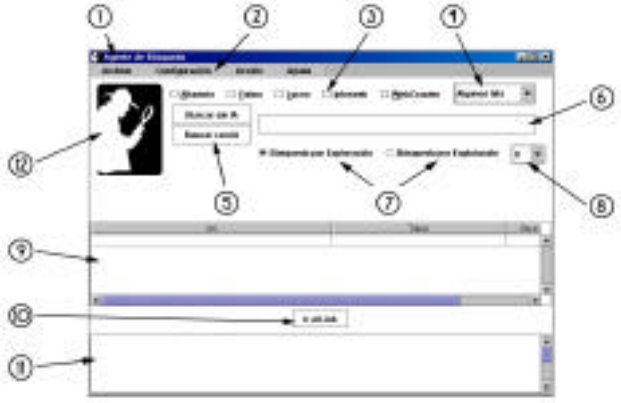

Figure 2 - User Interface of POIROT

Given a set of documents, the TF/IDF module computes and assigns the weights to each word to reflect the relative importance within the document and the set.

The ILP module generates the rules according to the weights computed and also assigns the relevance factor to the links returned. This relevance factor is used to expand the queries, rank the results and present them in that order to the user.

Results

We performed experiments with several user profiles of at least 20 bookmarks of specific subjects. The search engine used for comparisons was MetaCrawler.

In the first experiment we explore the capacity of the agent to find relations between relevant subjects. The second experiment was designed to verify the generalization ability of the agent, when the user does not use keywords that match exactly with the relevant subjects. Finally the third experiment presented test the capacity of the agent to adapt to changing interests.

In each experiment we report the **hits returned**, that is a good measure of coverage; the **position of the first relevant link**, that measure how good is the agent to rank the results and the **percentage of relevant links in the first 20 links** returned, that intend to give an idea of the precision of the agent.

User interested in "submarine tourism"

When a user with 23 bookmarks related with scuba diving and tourism, performs a query with the keyword: "diving" the results obtained by POIROT and MetaCrawler are presented in table 1.

The coverage, relevance and precision of POIROT outperform MetaCrawler. The difference in the percentage of relevant links in the first 20 links is 45%, this maximize the probability to find a good link in the first page returned by the agent. In the first 20 links POIROT returned just 1 link that was not related with the user interests.

	POIROT	MetaCrawler
Hits returned	413	70
Position of the first relevant link	1	3
Percentage of relevant links in the first 20 links	95%	50%

Table 1 User interested in "submarine tourism"

User interested in "Salvador Dalí"

In this case, using a profile with 21 bookmarks about Dalí, the user submits the query: "Surrealism", which is related with the subject of interest, but is not included in the subject description. The results are shown in table 2. Again the first and more that 50% of the first 20 links returned by POIROT were relevant.

	POIROT	MetaCrawler
Hits returned	407	72
Position of the first relevant link	1	10
Percentage of relevant links in the first 20 links	55%	10%

Table 2 User interested in "Salvador Dalí"

User interested in "Jaguars"

This experiment was divided in 3 parts, using the ambiguity associated with the subject "jaguar"[1]. In the first part the user was interested in Jaguars, the cats; in the second part the user was changing interests from cats to cars and in the third part the user is totally interested in cars.

[1] The idea of this experiment was taken from (Baeza 1998).

"Jaguar" is a word related with several subjects, such as cars, video game consoles, a football team and a pop group. The agent was able to resolve the ambiguity using the context created by the rules inferred.

With a profile consisting in 43 bookmarks related with jaguar cats, the results of issuing the query: "jaguar" are presented in table 3. POIROT expanded the query to "jaguar nature". The results are impressive in terms of relevance and precision.

	POIROT	MetaCrawler
Hits returned	421	67
Position of the first relevant link	1	43
Percentage of relevant links in the first 20 links	95%	0%

Table 3 - User interested in "Jaguar, the cat"

Using a profile of 43 bookmarks related to cats and 20 related to cars, simulating a progressive changing of interests from cats to cars and issuing the same query: "jaguar", the results are shown in table 4.

This time POIROT expanded the query to "jaguar haynes workshop usually manual sports british pubns crowood motorbooks bentley hours complete robert international edition". This expansion, apparently meaningless, captures the keywords present in web pages relevant to jaguar cats and cars.

	POIROT	MetaCrawler
Hits returned	315	67
Position of the first relevant link (cats)	3	43
Position of the first relevant link (cars)	1	2
Percentage of relevant links in the first 20 links (cats)	30%	0%
Percentage of relevant links in the first 20 links (cars)	55%	85%

Table 4 - User interested in "Jaguars, cats and cars"

The results are clear, showing how POIROT distributed the links presented between the two subjects, according to the interest shift of the user. MetaCrawler was victim once again of its lack of knowledge about user interests.

Note that the number of pages related to jaguar cars is considerably higher than the related to cats; that explain that the first link reported by POIROT was related to cars and the third to cats.

In the last step of this experiment, the user changed totally and now he is interested in jaguar cars. The results are presented in table 5

	POIROT	MetaCrawler
Hits returned	404	67
Position of the first relevant link	1	2
Percentage of relevant links in the first 20 links	95%	85%

Table 5 - User interested in "Jaguar, the car"

These results are consistent with the previous one; but what is curious is that, even in this case, MetaCrawler didn't report a relevant link in the first position, and the first link reported correspond to a Football team, the "Jacksonville Jaguars". This experiment illustrates the adaptive capacity of POIROT.

Discussion

According to the results, its clear that POIROT is able to:

- Learn relations between the subjects of interest to the user.
- Identify subjects related to the user interests.
- Resolve context issues regarding queries using the user profile.
- Adapt to interest shifts

The bookmarks proved to serve as a good and relative easy to access source of information regarding user interests, compared with the use of other sources as the local or proxy caches or the monitoring of user actions.

The extraction of subjects related to web pages, performed with a combination of the TF-IDF computation and a weighting based on the hierarchy of the html tags, was successful and allowed to characterize users interests and learn what was relevant to the user.

We didn't pay attention to the time required by the agent to perform the search. The time is generally higher than the needed by MetaCrawler, but given the percentage of relevant links in the first 20 links is very high, there is no need to navigate through several results pages or to manually improve the search to find the desired information.

The use of dictionaries and Thesaurus would probably improve the performance of the agent, allowing identifying synonyms, derivations, useful in the query expansion rule generation and multi-language support.

Another important aspect is the exploration of alternative sources of information about user interests. In particular we are interest in the monitoring of users actions such as follow a link, save or print a web page, etc. We will explore these issues in a forthcoming article.

References

Abrams, David 1997, *Human Factors of Personal Web Information Spaces*, Thesis University of Toronto.

Abrams, David; Baecker, Ron 1999, *How People Use WWW Bookmarks*, http://www.acm.org/turing/sigs/sigchi/chi97/proceedings/short-talk/da.htm

Baeza, Ricardo 1998. *Searching the World Wide Web: Challenges and Partial Solutions*. In Progress in Artificial Intelligence – Proceedings of Iberamia 98.

Bergadano, Francesco; Gunetti, Daniele 1995, *Inductive Logic Programming – From Machine Learning to Software Engineering*, The MIT Press.

Burges, Christopher 1998, *A Tutorial on Support Vector Machines for Pattern Recognition* Bell Laboratories, Lucent Technologies.

Cambridge Scientific Abstracts Stopword List, www.csa.com/stoplist.html

Chen, L.; Sycara, K. 1997, *WebMate, A Personal Agent for Browsing and Searching*, The Robotics Institute, Carnegie Mellon University.

Haykin, Simon 1999, *Neural Networks – A Comprehensive Foundation*, Prentice Hall.

Hock, Randolph 1999, *The Extreme Searcher's Guide to Web Search Engines*, CyberAge Books.

Joachims, Thorsten 1998, *Text Categorization with Support Vector Machines: Learning with many Relevant Features*. Fachbereich Informatik, Universität Dortmund.

Lawrence, S.; Giles, C.L. 1998, Searching the world wide web. *Science*, 280(5360):98.

Lawrence S.; Giles, C.L. 1999. Accessibility of Information On The Web. *Nature*, 400; 8.

Library of Congress 1999, *stopwords*, thomas.loc.gov/home/stopwords.html

Newell, Sima 1997, *Improved Internet Information Retrieval Through the Use of User Models, Filtering Agents and a Knowledge-Based System*, Thesis, McGill University.

Nilsson, Nils J. 1999, *Introduction to Machine Learning* (Draft version), Department of Computer Science, Stanford University.

Osuna, Edgar; Freund, Robert; Girosi, Federico 1997, *Support Vector Machines: Training and Applications*, Technical Report MIT Center for Biological and Computational Learning.

Pontil, Massimiliano; Verri, Alessandro 1997, *Properties of Support Vector Machines*, MIT Artificial Intelligence Laboratory Memo nº1612 and Center for Biological and Computational Learning Department of Brain and Cognitive Sciences Paper nº 152.

Salton G., McGill, M. G. 1983, *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

Search Engines: Education, Information and Great Links. people.delphi.com/ulialex/search_engines

Sonnenreich, Wes; Macinta, Tim 1998, *Guide to Search Engines*, John Wiley & Sons.

Sullivan, Danny ed. 1999a. How Search Engines Work and Search Engine Features Chart, www.searchenginewatch.com/

Sullivan, Danny ed. 1999b, The Search Engine Report 33, http://www.searchenginewatch.com/sereport/index.html

Tauscher, Linda; Greenberg, Saul 1999, Revisitation Patterns in World Wide Web Navigation, www.acm.org/sigchi/chi97/proceedings/paper/sg.htm