The Effects of Negative Interaction Feedback in a Web Navigation Assistant

Marcelo G. Armentano and Analía A. Amandi

ISISTAN Research Institute (CONICET/UNICEN)
Campus Universitario, Paraje Arroyo Seco, Tandil (7000), Argentina
{marcelo.armentano,analia.amandi}@isistan.unicen.edu.ar

Abstract. Recommender systems are a common solution used to assist users in searching and retrieving information on the web due to the benefits that can be obtained from the evaluation and filtering of the vast amount of information available. This article presents a user study on the feasibility of using negative interaction, that is the absence of interaction with some items in a list of suggestions, as implicit feedback used to improve the performance of a web navigation assistant. Results showed an increment of 16.65% in the acceptance of the suggestions provided by the assistant and an increment of 43.05% in the average use of the suggestions window when using negative interaction with respect to not using this feedback mechanism.

Keywords: Intelligent Agents, Web Navigation Assistance, Implicit Feedback.

1 Introduction

The Web is daily queried with multiple purposes, such as reading the news, researching about a specific topic, planning vacations or searching for references to answer specific questions. Search engines facilitate finding information quickly on any topic of interest. Using a set of keywords as an input, search engines offer as a result a list of Web links hopefully addressing the issues described by those keywords.

Search engines offer a quick access to the information indexed from the web, automating the classification of the contents to deliver the most relevant information to each query. However, it is unusual that users are able find the information they intend to in their first search because they are constantly overloaded with content and links. This can be frustrating to users, who know that the information is somewhere available but they are not able to find it. A usual problem with search engines is that many users do not know how to "build" the search queries to obtain the information they are looking for. The queries used in a search engine are usually short and ambiguous, and different users might use the same query with completely different needs [1].

To address this problem we propose a web navigation assistant that recommends possible interesting web pages to the user while navigating the web. The assistant observes the user interaction with a web browser to obtain implicit indicators for the subject of the user's interests. Then, the assistant automatically performs new web

M. Kurosu (Ed.): Human-Computer Interaction, Part III, HCII 2013, LNCS 8006, pp. 107–116, 2013.

searches to obtain pages related to those recently visited by the user. The interaction of the navigation assistant with the user is only through a small button indicating the availability of suggestions. When the user presses this button the top ranked results are shown in a window integrated with the web browser so that the user can continue navigating as usual if he/she is not interested in the suggestions.

We performed a set of experiments with real users in which we compared the results obtained considering the impact of two mechanisms of implicit feedback: the time spent on a page and the negative interaction. For these experiments we computed the percentage of usage of the window presented by the assistant, and the percentage of acceptance of the suggestions selected by the user. The results showed that considering both implicit mechanisms of feedback improve the performance of the assistant. Particularly, considering the negative interaction feedback allows the agent to modify his state when it has not properly inferred the user's search intention.

The rest of this paper is organized as follows. Section 2 presents some related work. Section 3 presents the web navigation assistant implemented, detailing its general architecture and how it process the implicit user feedback to improve the user profile for a searching session. Section 4 presents the user study conducted. Finally, Section 5 presents our conclusions of the study.

2 Related Work

There have been many research efforts focused on systems that recommend pages to web users. In this section we summarize some of those studies centered on obtaining implicit feedback from the user.

Our web navigation assistant learns a short term user model by analyzing the text of the pages that it assumes the user is interested in. Chi et al [2] obtains information about the user from the text of the links he/she clicks. However, many times the text of the links does not have relevant information about the linked page (e.g. "click here"). Zhu et al. [3] build a user model by analyzing the user's behavior in order to build rules such as "any word which is present in three consecutive pages visited by the user will be present in a page that is important to the user". These rules, called "abstract navigation patterns", are generated according characteristics of web pages visited by the user and not with any particular page. Matthijs and Radlinski [4] presented a personalization approach that combines the content and previously visited websites to build a user interest profile using the users' complete browsing behavior. This model is then used to re-rank web results. Hu et al. [5] also proposed a personalization approach but restricted to searching web services. They obtain the user interests from both the search requests and the previously used services by extracting the meaningful words from the service request and the used service description files.

The time spent in a page has been commonly used in many approaches as an indicator of the user interest in any piece of information. Parsons et al. [4], for example, found that there is a positive relationship between the time users spend watching an item available for purchase and their intention to buy it. Although the time spent watching an item depends also of external factors (such as the amount of visual details, images, distractors, etc.), it is believed that it is a potential indicator of the user preferences.

Joachims et al. [7] studied the behavior of the users interacting with the results provided by a search engine. The results obtained shown that users take decisions from the descriptions provided by the search engine, by clicking on the links they are interested. However, the user clicks are conditioned by the order in which the links are shown. The user mainly interacted with the top ranked results (that is the links in the top of the list) even when the descriptions were less relevant that results in lower positions. Moreover, the user spent more time reading the descriptions of the top ranked results, and descriptions at lower positions received less attention.

Our approach takes the findings of Joachims et al. [7] to define the concept of negative interaction, that is to use the information of the search results skipped by the user to improve the user profile related to his/her information needs in a search session.

3 Web Navigation Assistance

WebHelper is a web navigation assistant we developed to observe the user behavior while interacting with a web browser to detect his/her subject of interest and suggest potentially interest pages. The objective of the assistant is to help the user to find web pages that he/she might not be able to find with the keywords he/she is using.

The search intention of the user is inferred by processing the text of the web pages he/she visit. After each visited page, the assistant updates the weights assigned to different words that describe the search intention of the user in a session. The top ranked words are then used as keywords to perform a new web search and present the results obtained to the user in an independent window integrated with the browser. This window is only shown when the assistant has enough information to infer user's the search intent. However, the user is able to close this window at any time and to open it asking for suggestions.

3.1 General Architecture

WebHelper is implemented server-side, in the form of a web proxy. This way, the user can start using the assistant simply configuring his/her web browser to use the assistant's proxy. When the user sets his/her browser to use this proxy, all requests are intercepted and two processes are applied. The first process extracts the information contained in the response pages and assigns different weights according to the context of each extracted term. The second process injects in the HTML response the code corresponding to the suggestion window, if appropriate.

The dataflow in *WebHelper* is shown in Fig 1. The user searches the internet using a web browser. The *WebProcessor* module captures the response to the server request and executes two processes: information extraction and information injection. The information extraction process captures the content of the web page and passes it to the *WordProcessor* module. If the assistant has any recommendation to present to the user, the information injection process adds to the response the code corresponding to the suggestions window to be presented to the user. On the other hand, the *WordProcessor* module takes the page content received from the *WebProcessor* module and updates the user profile. We describe this process in Section 3.2

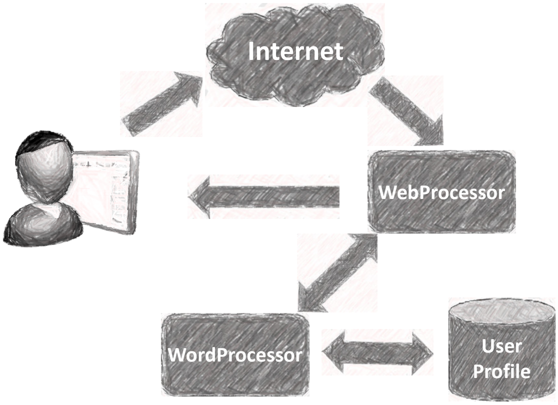

Fig. 1. WebHelper general architecture

3.2 Information Processing

When the *WordProcessor* component receives a new visited document, it first converts it to plain text, detects its language[1], removes stop words and applies the corresponding Porter stemming algorithm [10].

The user profile keeps the set of most frequent words, each of them with an associated weight. These weights vary according to the place in the page in which each word appears, and vary after each visited page, either by adding or subtracting a new constant value. Different constant values were defined for words appearing in headings, normal text, links, text typed in input boxes, meta-information, etc.

On the other hand, words weights in the user profile are decremented for three different reasons: (1) the word does not appear in the current page; (2) the negative interaction feedback is applied; (3) the user does not select any suggestion after a fixed amount of visited pages. We give details about the last two items in Section 3.3.

After processing each visited page, the following situations can arise:

- There were no suggestions up to that moment, and no terms in the user profile exceed the “suggest-me” threshold. In this case, nothing happens and no suggestions are given to the user.
- There were no suggestions up to that moment, and two or more words exceed the “suggest-me” threshold. In this case, the assistant use the words exceeding the threshold and performs a web search using those words as keywords. The top results of the search engine are presented as suggestions. In the current implementation of *WebHelper*, the Google® API is used for querying the web.

[1] We currently work with English and Spanish

- The assistant was already showing suggestions, but after processing the information of the new document the weights of all words are under the "suggest-me" threshold. In this case, the suggestions window is not injected in the response to the user.
- The assistant was already showing suggestions and after processing the information of the new document the words whose weights exceed the "suggest-me" threshold are the same that in the previous cycle. In this case, suggestions are rotated, and a new set of web pages from the previous query are shown in the suggestions window.
- The assistant was already showing suggestions and after processing the information of the new document a new set of words weights exceed the "suggest-me" threshold. In this case a new web search is performed using those new words as keywords and the top results are presented as suggestions, replacing the previous window.

Before querying the web, the Google® API is used to obtain suggestions for each candidate keyword. This service provided by the Google® API, receives a word as input and returns a set of related words as output. The suggested words for each keyword and the keywords themselves are combined to query the search engine. This step improved the performance of the assistant since the terms of in the user profile are words trimmed by the stemming process. Using those stems to query the search engine produced unexpected results.

3.3 Feedback Processing

To determine the search intention of the user in a session, the navigation assistant considers two implicit feedback indicators: the time the user spends in a page and the interaction with the suggestions window.

The time the user spends on a page have been extensively used as an indicator of interest in a piece of information [4][8]. A web page is considered active by *WebHelper* when the user first access to it or when it is selected by using the browser's tabs.

We consider two types of interactions with the suggestions window: positive and negative interactions. A positive interaction is triggered when the user clicks on link in the suggestions window. In this case, the content of the suggestion (title and snippet) is used to improve the user profile for that session.

Moreover, at the same moment when the user shows interest in a suggestion, there is also a negative interaction with respect to the items located above the one selected by the user in the list of suggestions. We apply a heuristic based on statistical studies that showed that lists are usually read from top to bottom [7]. This way when the user selects an item in a list, he/she is not only indicating interest in the suggestion selected but also is implicitly indicating that he/she is not interested on the previous items. Assume for example that the assistant presented a list of suggestions <s1, s2, s3, s4, s5, s6, s7> and the user clicked on s1, s3 and s5. We can assume that s3 is more relevant that s2, since the user read the description of s2, but preferred clicking on s3.

For the same reason we can assume that s5 is more relevant that s2 and s4. We can assume that s2 and s4 are not relevant to the user's needs and we can use the information they contain to update the user profile as negative information [9]. The suggestions list can then be re-organized to eliminate the preceding recommendations.

Similarly, if the user has not interacted with the suggestions window after a certain amount of visited pages, it is assumed that the suggestions are not interesting to the user. Consequently, suggestions are rotated to show the user a new set of possible interesting websites. Following the idea of negative interaction, when suggestions are rotated the weights of the terms appearing in the hidden suggestions information are decreased.

3.4 Suggestions Window

When the assistant has suggestions to make to the user, it creates a drop down window integrated to the page returned to the user. This window is situated on the right of the returned page and contains a maximum of eight suggestions, consisting in a title and a snippet (Fig. 2). This window can be hidden and re-opened at any time.

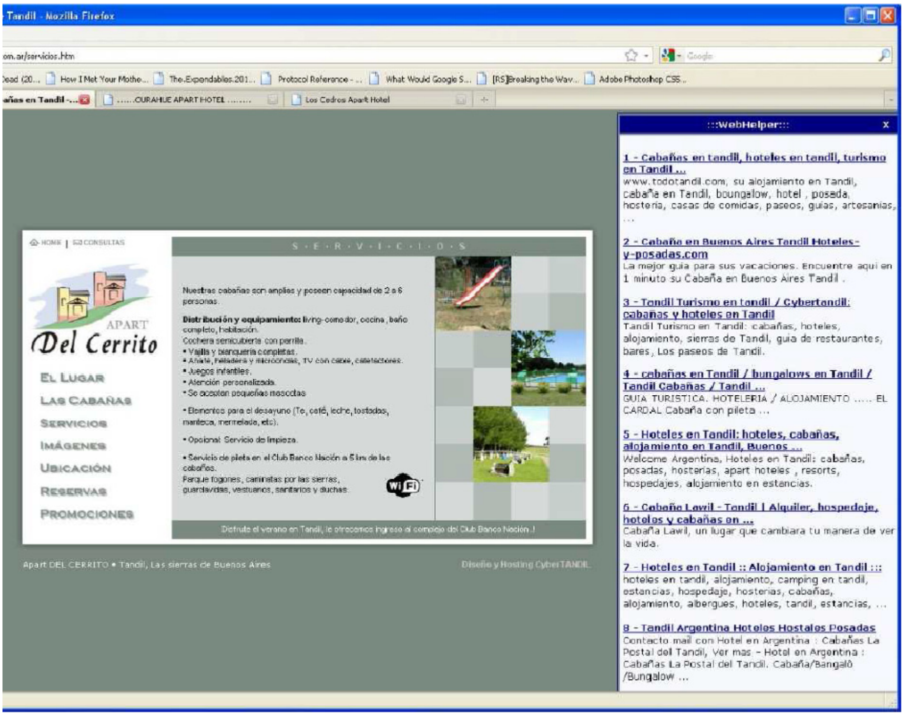

Fig. 2. Suggestions window while visiting a page

4 Experimental Evaluation

4.1 Experiment Setup

To validate the proposed approach, we tested our system with 32 volunteer users, both male and female, with ages ranging 18 to 55, and different levels of expertise in searching the web. Users were divided into groups to test four different configurations of the system.

We asked each user to freely use the web to search for information about any subject of interest. The navigation assistant was slightly modified for these experiments: after selecting a suggested site from the suggestions' window, users were asked whether the page they had just visited was interesting according to their search intentions. This feedback was only logged to compute the performance of the assistant and it was not used to modify its normal behavior since in a live scenario feedback is never asked to users.

The duration of the experiments was, on average, 25 minutes. Most users started their session accessing Google's search engine and entered the keywords they believed best expressed their information needs. Other users interested in news opened their preferred online newspaper and start browsing through it. Finally, other users visited pages they usually visit.

4.2 Metrics

Two metrics were used to evaluate the performance of our navigation assistance: the percentage of use of the suggestions' window and the percentage of acceptance of the suggestions selected by the user.

To compute the average use of the suggestions' window, we divided the number of suggestion selected by the user by the number of windows shown to the user during the experiment.

$$use\_of\_the\_assis\tan t = \frac{\#suggestions\_selected}{\#suggestions\_shown}$$

Likewise, the percentage of acceptance of the suggestions is computed dividing the average number of accepted suggestions by the average number of suggestions selected by the user.

$$accep\tan ce = \frac{\#suggestions\_accepted}{\#suggestions\_selected}$$

4.3 Time Spent on a Page

The first experiment explored whether considering the time the user spent on a page any influence the performance of the assistant. In the first configuration of the navigation assistant, any page visited by the user was processed, without considering how long the user remained in the same page. We use this first configuration as a base-line for comparison with the other three configurations. In the second configuration, pages were only considered for improving the user profile when the user spent more than a predefined amount of time reading the page. For our experiments this threshold was set to six seconds. In this configuration, the system took more time to generate suggestions since not all information visited by the user is processed and, consequently, the weights assigned to individual terms increased more slowly than in the first configuration.

Since different experiments took different durations, the average results were normalized by the duration of the individual experiment. For example, nine suggestions windows were shown to User 1 for the first configuration, and User 1 used the system for 30 minutes, so we used a value of 0.3 windows per minute. Table 1 shows the results obtained.

Table 1. Results for user tests with and without time processing

	Suggestion windows shown	*Suggestions selected by the user*	*Suggestions accepted by the user*	*Time for showing the first suggestions*	*Duration of the experiment*	*Google searches*
Without Time processing	0.662 per minute	0.148 per minute	0.113 per minute	3.62 minutes	22.125 minutes	0.414 per minute
With time processing	0.350 per minute	0.184 per minute	0.163 per minute	10.62 minutes	22.375 minutes	0.373 per minute

From Table 1 we can compute that the average use of the assistant without considering the time spent on a page was 22.35% while the average acceptance of the suggestions selected by the user was 76.35%. On the other hand, when considering the time the user spent on a page, these values are increased to 52.57% and 88.59% respectively.

4.4 Negative Interaction Feedback

The third configuration of the experiment was set up to consider negative interaction feedback but did not consider the time spent on a page. Finally, the fourth configuration used both feedback mechanisms together. Table 2 shows the results obtained. A new column is added to show the number of times that the suggestions were rotated.

The average acceptance of the suggestions selected by the user was 87.1% (with an average use of 26.59%) if we consider the negative interaction feedback alone. However, we reach an average acceptance of 93% and an average use of 65.4% by combining both feedback mechanisms.

Table 2. Results for user tests with and without time processing

	Sugges-tion windows shown	*Sugges-tions selected by the user*	*Sugges-tions accepted by the user*	*Time for showing the first sugges-tions*	*Duration of the experi-ment*	*Google searche s*	*Rotation of sug-gestions*
Without Time pro-cessing	0.583 per minute	0.155 per minute	0.135 per minute	8.750 minutes	24.000 minutes	0.307 per minute	0.102 per minute
With time pro-cessing	0.327 per minute	0.214 per minute	0.199 per minute	10.500 minutes	22.625 minutes	0.140 per minute	0.072 per minute

4.5 Discussion

For the first configuration, while the acceptance of the suggestions selected by the user was high (76.35%), the use of the suggestions window was quite low (22.35%). This fact suggests that the information shown to the user in the suggestions' window might have not convinced the user to click on a suggestion. Since for this experiment all visited pages were processed, the threshold for showing suggestions was reached quickly, but the system was not able to properly infer the intention search of the user. The second experiment solved this problem by increasing the use of the suggestions window to 56.57% at the expense of the time that the system needed to give assistance to the user.

On the other hand, by using the negative interaction feedback alone, we obtained an acceptance of the suggestions similar to that we obtained when we used the time processing alone. The average usage in this case was better than the first configuration but lower than the second configuration. The time spent to show suggestions to the user is lower than the second configuration, but higher than the first configuration. This is due to the fact that the negative interaction lowers the weights of the terms that are present in the unvisited documents. Then these weights trend to increase slowly.

By combining both feedback mechanisms we obtained a better general performance of the assistant. Furthermore, although the time that took the system to show the first suggestions window was similar to that in the second configuration, it performed fewer queries to the search engine, achieving a better acceptance of the suggested sites.

5 Conclusions

In this article we described a web navigation assistant that infers the search intention of a user in order to provide him with suggestions of sites in which he/she might be interested. The assistant uses implicit feedback mechanisms to improve a temporal user profile that contains information about the subject of interest in a search session.

We presented a user study on the feasibility of using negative interaction, which is the absence of interaction with some items in a list of suggestions, as implicit feedback used to improve the performance of the web navigation assistant.

The study allowed us to conclude that using both the time spent on a page and the negative interaction feedback improves the performance of the assistant, obtaining an increment of 16.65% in the acceptance of the suggestions provided and an increment of 43.05% in the average use of the suggestions window.

References

1. Hu, R., Dou, W., Liu, X.F., Liu, J.: Personalized Searching for Web Service Using User Interests. In: Proceedings of IEEE Ninth International Conference on the Dependable, Autonomic and Secure Computing (DASC), pp. 156–163 (2011)
2. Chi, E., Pirolli, P., Chen, K., Pitkow, J.: Using information scent to model user information needs and actions and the Web. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 490–497. ACM, New York (2001)
3. Zhu, T., Greiner, R., Häubl, G.: Learning a model of a web user's interests. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 65–75. Springer, Heidelberg (2003)
4. Matthijs, N., Radlinski, F.: Personalizing web search using long term browsing history. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011), pp. 25–34. ACM, New York (2011)
5. Hu, R., Wanchun, D., Liu, X., Liu, J.: Personalized Searching for Web Service Using User Interests. In: Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC 2011), pp. 156–163. IEEE Computer Society, Washington, DC (2011)
6. Parsons, J., Ralph, P., Gallagher, K.: Using Viewing Time to Infer User Preference in Recommender Systems. In: Proceedings of the AAAI Workshop on Semantic Web Personalization held in conjunction with the 9th National Conference on Artificial Intelligence (2004)
7. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005), pp. 154–161 (2005)
8. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. Commununications of ACM 40(3), 77–87 (1997)
9. Kurosu, M., Ookawa, Y.: Effects of Negative Information on Acquiring Procedural Knowledge. In: Proceedings of the International Conference on Computers in Education (ICCE 2002), p. 1371 (2002)
10. Porter, M.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)