Keyword Extraction Based on Implicit Feedback

Jakub Kříž[*]
Institute of Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
jacob.kriz@gmail.com

Abstract

To improve the results from search engines and make them more personalized for the user, we need to find out about the interests of a particular user. Many of the search personalization methods analyse documents visited by the user and from these documents infer the user's interests. However, this approach is not accurate, because the user is rarely interested in the whole document; he might be interested in parts of the document only or the document does not have to interest him at all. In this article we analyse the user's activity on a web site, called implicit feedback. This feedback is represented by the user's behaviour in the Web browser: time spent browsing, mouse cursor movement, clicking, scrolling etc. Our method is then able to more accurately extract keywords from the documents by putting more weight on the keywords the user is more interested in and disregarding those which did not interest him. We extract keywords directly from the text and also by using traditional methods enhanced by the implicit feedback.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*

Keywords

implicit feedback, keyword extraction, user model

1. Introduction

On the internet we can find a vast amount of information, usually organized into documents. To look for a particular document we can use full text search engines, which are very popular nowadays. These search engines crawl trough the documents regularly and can quickly find all the documents which contain the terms we are looking for. The number of documents containing the terms is usually very large and they have to be sorted based on their relevance to the keywords and their importance, so that we can quickly find the right documents.

However, the search results can be often unsatisfying for the user; multiple words may have similar meanings (synonymy) and words can have more than one meaning (polysemy). This can cause the results being different than expected, especially when searching using a lower number of keywords [7].

For this reason it is better to personalize the search results for the needs of a particular user. To make this possible, we need to find out about the interests of the user and create a user model. This user model often consists of the metadata extracted from documents the user has visited, because of the assumption, that their content represents the user's interests.

This assumption is not completely correct. The user's behaviour on the web can be very erratic and the user rarely reads an entire document he loaded. Most of the time, he is only interested in parts of the document or it might not interest him at all in case he stumbled upon the document by chance or the source which has lead him to the document was misleading. To make the user model more accurate, we need to know which of the visited documents actually interested the user and, ideally, which parts of the documents.

We can find out about the user's interest in the document – feedback in two ways: explicitly and implicitly. If we want to collect the feedback explicitly, we need to ask the user for some additional input. This is impractical and time consuming for the user and can be even inaccurate, because the user's true interests might not be well represented by his explicit feedback. Having to stop to enter explicit ratings can alter the user's behaviour and the users read a lot more articles than they rate [2].

On the other hand, implicit feedback can be collected all the time and in the background without the need to bother the user. The main disadvantage of implicit feed-

[*]Bachelor study programme in field Informatics. Supervisor: Tomáš Kramár, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, STU in Bratislava.
Work described in this paper was presented at the 8th Student Research Conference in Informatics and Information Technologies IIT.SRC 2012.

Kříž, J. Keyword Extraction Based on Implicit Feedback. Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Student Research in Informatics and Information Technologies, Vol. 4, No. 2 (2012) 43-46

back is that it is more difficult to process with reliable results. In this work we aim to design a method for collecting and processing implicit feedback on the web.

2. Implicit Interest Indicators

The collection of implicit feedback on the web is limited by the web browsers. We cannot simply observe any factor we might need – for example, to observe the user's gaze we need additional hardware and/or software, which majority of users do not have. We have to focus on the indicators which can be collected in all widely used browsers using a client-side scripting language.

The indicators we can observe are based on standard input methods – keyboard and mouse and on the time spent browsing. The indicators, when detected, represent user's interest in the document. These can be divided into two main groups: local and global.

2.1 Local Indicators

Local indicators can be used to determine which parts of the document was the user interested in. A rule of thumb which can be observed in related works [4, 5] says that the less the indicator occurs, the more reliable and accurate it is. This reliability cannot be exactly quantified, but we can estimate it based on the experiments conducted by other works.

The indicators we observe follow roughly ordered by the reliability:

Text copying
Text selection and copying is considered to be a very strong indicator of interest in the targeted text [4, 5, 6]. The additional advantage of this indicator is that we can extract the targeted text very accurately.

Text selection
Text selection is less reliable than copying but still a very strong indicator of interest [4, 5]. Some users keep selecting the text they are currently reading, which is good for detecting which parts of the document the users have read, however, in this case it can no longer be used to extract the keywords directly.

Text clicking
Clicking on the parts of the text is similarly accurate as selection and usually occurs more often [4].

Text tracing
Some users use the mouse cursor as a helping device while reading the text, when they move the cursor over or under the line of text that they are currently reading [5].

Distance from the moving mouse cursor
Experimental data show that the user's gaze is closer to the mouse cursor when the cursor is moving [1, 4, 8]. The data also show that this indicator is more reliable when the user is in general more active with mouse movement [4, 9].

Distance from the stationary mouse cursor
Even when the mouse cursor is stationary we can use its position to determine the user's gaze, although with less certainty. A specific usage of the mouse cursor, which helps our cause, is to mark an interesting link or part of the text by putting a cursor close by [11, 9].

Time spent on the screen
Generally speaking, the user is more interested in the content in the middle of the web browser window [4] and the more time the content spends on the screen, the more likely is the user interested in it [6], especially when the user is actively reading the document [10].

2.2 Global Indicators

Global indicators can show that the entire document is interesting for the user, but we are not able to determine which part of the document is the most interesting when we detect them. The indicators we observe are following:

Time spent scrolling
Experimental data shows that users tend to scroll the web page for a longer time when they are potentially interested in it [2].

Saving, bookmarking and printing the web page
These are all very strong indicators [5], however, it is difficult to detect them trough client-side scripts. We can at least detect the keyboard short cuts used for these tasks, although that makes them occur even more rarely.

Typing into a form
When the user types on a web page we can assume it has captured his interest. For example typing a comment to an article means that the user finds the article interesting, but we are not able to find out which part of the article motivated him to write the comment.

3. Processing the Indicators

The goal of this process is to extract keywords from the document and rate them in a way which is representative of the relevance of the keywords as well as the user's interest in them.

The method we use to process the indicators is based on the methods used in related works. Hijikata [5] used some of local indicators to directly extract keywords from the document. Hauger et al. [4] used local indicators detected over paragraphs of text to determine how much time the user has spent reading the particular paragraph.

Our method combines these two methods and adds global indicators to determine the user's overall interest of the web page. We use local indicators to determine how interested the user was in the paragraph and then use the tf-idf method [3] to extract keywords from them. Tf-idf is a fundamental method used for this purpose and it is well known to return accurate results. In addition to keywords extracted by tf-idf we use the most precise local indicators to directly extract keywords from text.

The algorithm works in the following way:

1. We break down the document into text elements based on its HTML structure, usually into paragraphs.

2. Each element is given the rating of 0 to begin with.

3. When a local indicator (described in the previous Section) is detected:

   (a) The rating of the element which was targeted by the indicator is increased. The rating is increased by a larger value when the indicator is considered more reliable and accurate. This increment is based on the experiments done by related works and our own.

   (b) If the indicator was one of the two most accurate, which are *text copying* and *text selection*, we extract the exact words from the area where they appeared. In case the extracted text is short and consists of a few words only, we can consider them keywords without any further need for extraction.

4. The rating of all the elements of the document is increased when there is a global indicator detected.

5. When the user leaves the document:

   (a) We compute the rating of the entire document by adding the ratings of its elements.

   (b) We compute the relative importance of each element – its rating compared to ratings of other elements of the document

   (c) From each element we extract keywords using the tf-idf method.

In the end, every element will have a relative rating RR_E calculated in the following way, where N_i is the number of indicators detected for the element, I_{S_i} is the general importance of the particular indicator, I_{U_i} is the importance of the indicator for the particular user and N_e is the number of elements of the document:

$$R_E = \sum_{i=1}^{N_i} I_{S_i} I_{U_i}$$

$$RR_E = \frac{R_E}{\sum_{i=1}^{N_e} R_{E_i}}$$

An important part of the method is detection of the user's inactivity. We do this based on a few key factors, namely *mouse movement and clicking* and *keyboard input.* These factors were chosen because their absence clearly shows the user's inactivity. If we do not detect these indicators for a period of time we consider the user inactive and stop incrementing the elements' ratings. The length of the period after which we consider the user inactive is based on the particular user – we incorporate the rating of user's average activity on all the web pages he visits into the user model and based on this value we compute the length.

4. Realization of the Method

To be able to track the user we use a personalized proxy server[1], which stands between the client and the server. Each request is first handled by the proxy server, which sends it further. The response is again first handled by our proxy server. At this point we can inject our scripts for collecting the user feedback to the web page and send it to the client.

The method which detects indicators and processes them is realized in JavaScript. The script runs on the background during the user's browsing, detects the indicators and rates the elements. When the user leaves the web page the script processes all the data and sends it via Ajax to the proxy server. On the proxy server we extract the keywords using the tf-idf method.

A problem we might run into is that some users do not leave the web page for a long time, they might just leave it open in the background of their web browser. This should happen rarely and to a very specific group of users, so we do not address this issue at the moment.

Table 1: General success of the direct keyword extraction.

method	yes	no
text copying	100%	0%
text selection	62%	38%
text tracing	33%	67%

5. Experimental Evaluation

As described in Section 3, our method extracts keywords in two ways: from paragraphs using the tf-idf method and directly from the text when one of the more accurate indicators is detected. In this Section we describe the evaluation of both methods.

The evaluation is based on comparing the implicit feedback from our method and the explicit feedback, which we asked the users to manually input. The participants of the experiment were instructed to browse the web as usual for a few hours and encouraged to enter reliable explicit feedback, which should eliminate possible discrepancies. When leaving each web page the user visited we presented to him a few keywords we obtained from his behaviour. His task is to rate the keywords "yes" or "no" if he feels like the keyword matches accurately his interests in the web page. The users did not know how either of the methods worked and a few random words from the page were put in the keywords to discourage the users from blindly selecting "yes".

5.1 Direct keyword extraction

First we evaluated the three separate methods for extracting keywords directly: text selection, text copying and text tracing. 10 users participated in this experiment. We looked at general success of the extraction of a relevant keyword – for a keyword we computed the percentage of the time it was considered correct. This is summarized in Table 1. For text selection and text tracing we looked at two other factors: the number of words extracted and the number of times the indicator was detected on the particular web page.

Text copying proved to be extremely accurate, however, it was detected very rarely. During our experiment the users only copied small amounts of text, so even though this is a very accurate indicator, we are not able to detect it often enough. Users selected text more often than copied it and they selected longer text segments. Interesting fact was that if the total number of selections on the web page was low, the users marked the keywords as accurate even from selections 10 words long and more. We can conclude that, after filtering out the stop words, we can use text selections as direct extraction indicator if the total number of selections was lower than 4.

[1] http://peweproxy.fiit.stuba.sk

Table 2: General success of the keyword extraction from elements.

method	elements	entire document	random
yes	108	91	22
no	44	55	124
success rate	71 %	62 %	15 %

Text tracing did not test well as a direct keyword extraction indicator and it does not seem to be useful even filtered it based on length or number of occurrences. This can be partially due to the difficulty of recognizing it via client scripts, which can sometimes mistake random mouse movements as tracing. Another reason might be the difficulty of extracting the exact text the trace was connected with. Finally, some users just use it as a reading aid and not an aid to mark significant keywords. Nevertheless, we can use the indicator to rank the paragraphs where it occurred and extract keywords from it via other methods, as described in the previous Sections.

5.2 Keyword extraction from elements

Evaluation of the second part of the method was done in a similar way. We extracted the keywords from a document using two methods: extraction from elements, extraction from the entire document and random keywords, which, in addition to using them to discourage the users from being biased we used them as a control group. The results of the evaluation are summed up in Table 2.

The success rate of the keyword extraction from elements was rather high which shows the ability of our method to extract keywords based on the detected implicit interest indicators. This rate was also higher than that of the keywords extracted from the entire document. Even though the latter method is very accurate, we were able to improve it.

The random keywords had a significantly lower success rate than the two evaluated methods.

Acknowledgements. This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

References

[1] M. C. Chen, J. R. Anderson, and M. H. Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *CHI '01 extended abstracts on Human factors in computing systems*, CHI EA '01, pages 281–282, New York, NY, USA, 2001. ACM.

[2] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, IUI '01, pages 33–40, New York, NY, USA, 2001. ACM.

[3] A. Goker and J. Davies. *Information Retrieval: Searching in the 21st Century.* Wiley, 1st edition, December 2009.

[4] D. Hauger, A. Paramythis, and S. Weibelzahl. Using browser interaction data to determine page reading behavior. In *Proceedings of the 19th international conference on User modeling, adaption, and personalization*, UMAP'11, pages 147–158, Berlin, Heidelberg, 2011. Springer-Verlag.

[5] Y. Hijikata. Implicit user profiling for on demand relevance feedback. In *Proceedings of the 9th international conference on Intelligent user interfaces*, IUI '04, pages 198–205, New York, NY, USA, 2004. ACM.

[6] M. Holub and M. Bielikova. Estimation of user interest in visited web page. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1111–1112, New York, NY, USA, 2010. ACM.

[7] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36:207–227, January 2000.

[8] M. Labaj. Implicit feedback based recommendation and collaboration. *Information Sciences and Technologies Bulletin of the ACM Slovakia*, 3:41–42, 2011. ACM.

[9] C.-C. Liu and C.-W. Chung. Detecting mouse movement with repeated visit patterns for retrieving noticed knowledge components on web pages. *IEICE - Trans. Inf. Syst.*, E90-D:1687–1696, October 2007.

[10] D. Oard and J. Kim. Implicit feedback for recommender systems. In *in Proceedings of the AAAI Workshop on Recommender Systems*, pages 81–83, 1998.

[11] K. Rodden, X. Fu, A. Aula, and I. Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 2997–3002, New York, NY, USA, 2008. ACM.