A CONTENT-BASED APPROACH FOR DETECTING USERS' SHIFT OF INTERESTS

Jean-Yves Delort[1]

[1]LIRMM – Montpellier 2 University, France
delort@lirmm.fr

ABSTRACT

Adaptive Navigation Systems (ANS) are intended to support search in hypermedia environment. On the Web, ANS cope with the lack of user feedbacks with unintrusive methods to detect and understand the users' searching behaviours. This article tackles to the problem of detecting Web users' shifts of interests. Most existing methods focus on implicit interest indicators like the time spent on a page or the amount of page read. However, previous researches on these indicators have shown they do not convey enough clues about the users' interests. This paper presents a new unintrusive method to detect the shifts of focus using the content of the accessed pages. The proposed approach is based on the assumption that the issue of detecting the shifts of focus comes down to detecting paragraph boundaries in a very large document made up with the content of the accessed documents. The article also shows how an existing shift detection algorithm can also be considered as a text-segmentation approach. The two approaches are compared with an evaluation with real-users.

KEYWORDS

Shifts of focus, implicit interests, user needs.

1. INTRODUCTION

Adaptive Navigation Systems (ANS) are intended to support the users during their navigations. In order to provide a relevant and useful support, they have to understand the user's behaviour. On the Web, ANS can usually not rely on the users to give their feedback. They have to unobtrusively track the clues about the users' needs and search strategies. This article addresses the issue of implicitly detecting the users' shifts of interests.

There is extensive literature on the characterization of the users' interests [1,2,3]. Most existing methods are based on interest indicators such as the time spent on a page, printing, or interactions like bookmarking, text highlighting, or scrolling. In [4] a model that implicitly learns the user's current interest using the content of the accessed documents was proposed. The model relies on the two following assumptions:

1. the content of the accessed documents contains terms related to the user's interest,
2. the pieces of information that are common to most pages during the same search activity are relevant clues about the user's current interests.

[4] also presented a technique to detect the user's shifts based on this model. To our knowledge, it is the first method to address the issue of detecting user shifts based on the accessed contents. However, although this technique has a good prediction rate, it requires a complex tuning.

In this paper we introduce a new approach to unobtrusively detect the user's shifts using the accessed contents. It is based on the assumption that the issue of detecting shifts of focus is isomorphic to the issue of detecting paragraph boundaries in a document made up with the previously accessed contents. The similarity between these two contexts is interesting because it implies that almost every text-segmentation algorithm is able to detect session boundaries.

The sequel of this article is organised as follow: In the next section we review related work on detecting users' shifts or session boundaries. In Section three we introduce our shift detection algorithm. It is based on TextTiling, a well-known text-segmentation algorithm. In Section four we show that the shift detection algorithm proposed in [4] can also be considered as a text-segmentation algorithm. In Section five we put forward an experimentation with real users intended to compare the two methods. We observe that both methods can efficiently detect the users' shifts but the text segmentation-based approach is more sensible than the clue-based approach while the latter has the highest specifity. Section six concludes and outlines the future work.

3. RELATED WORK

In this section we review existing techniques to detect changes in the user's profile or behaviour and we discuss how they differ from the issue of detecting users' shifts of interests.

[5] describes an agent that compiles a daily news program for individual user. It is based on a user model that distinguishes between the user's short-term and long-term interests. The model is able to predict the user's interests for incoming stories based on other stories that have previously been rated. The model considers that every new story that is ranked by the user corresponds to a novel short-term interest.

The issue of detecting the users' shifts can be compared to the issue of detecting the users' session boundaries. The difference comes from the concept of search activity: A search activity is the sequence of the user's interactions intended to satisfy an initial goal. A shift corresponds to an interaction that marks the beginning of a new search activity. A user session usually refers to the sequence of pages accessed by a user on the same website in order to achieve a task. Clearly, session boundaries do not necessarily correspond to users' shifts. Indeed, a user can access several times a website within the same search activity. Several articles have tackled to the issue of detecting the session boundaries. In [6,7,8] the proposed approaches are time-based. For example, [6] assumes that accessed pages within a given time span belong to the same session. In [8], pages belong to the same session if the gap of time between two consecutive pages is lower than a given threshold.

The issue of detecting the users' shifts can also be compared to the issue of detecting how the user's queries change. Changes between two consecutive queries are called search tactics. The difference stems from the flow of interactions considered. In order to detect search tactics, one relies on explicit information, the user queries, which are supposed to represent the user's needs. On the other hand, in order to detect the shifts of focus one usually cannot rely on explicit information given by the user. One must looks for the shifts within the interaction trails. In [9] seven search tactics are identified. They have designed a bayesian network which takes into account the previous search tactic and the amount of time since the last query occurred. The network is used to predict the next search tactic. They have also proposed an enhanced version of this network that takes into account the topic of the previous query.

4. Using Lexical-Cohesion To Detect Shifts

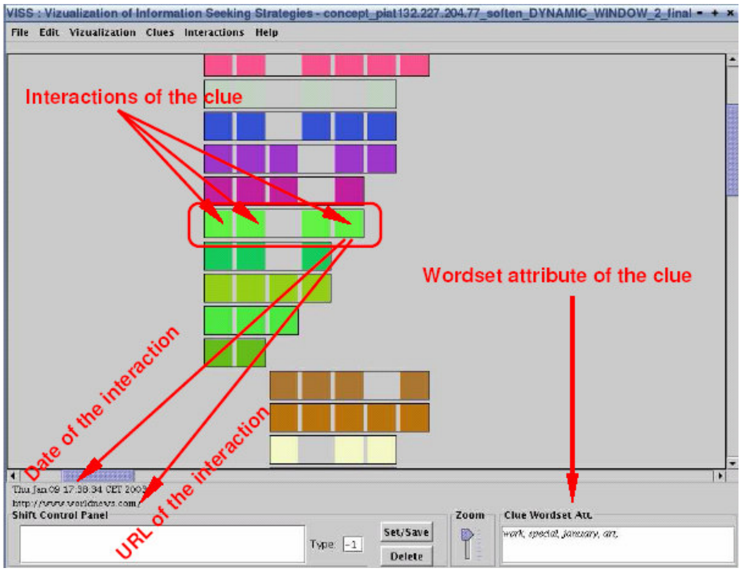
Text-segmentation techniques aim at detecting the paragraph boundaries in a document. In this section we explain how they can be used to detect the user shifts. Let us consider the "history" document made up with the content of all the previously accessed documents. We can represent a sentence in this document by the whole content of an accessed document. Accordingly, finding user shifts comes down to finding paragraph boundaries in this document. Paragraph boundaries in the "history" document are user shifts.

Many of text-segmentation techniques are based on lexical cohesion. Lexical cohesion approaches assume that sentences of related terms are likely to be related [10]. We have chosen a well-known text-segmentation method, TextTiling [11] mostly because it can detect paragraph boundaries on-the-fly and because it is easily tuneable. The following version of TextTiling is based on [12]:

1. First, the input text is divided into individual lexical units of predefined size *w*.
2. Then, adjacent pairs of blocks are compared. A similarity value (the cosine) is calculated at each interval of predetermined size *s* between the blocks of words on the right and on the left of the current position.
3. Finally, the algorithm deduces boundary points from these similarity values by supposing that high depth scores (major drops in similarity) indicate topic boundary points. Paragraphs boundaries can be deduced taking the nearest sentences of the detected boundary points.

4. Relationships Between The CEA And Lexical-Cohesion

In this section we bring out how the approach proposed in [4] can be connected to lexical cohesion. This approach is based on the Clue Extraction Algorithm (CEA). Details of the approach can be found in [4]. A clues is a pair (*I*,*W*) where *I* is a set of previously accessed pages and W is the set of common terms to the contents of the documents in *I*. *W* must be non-empty. The CEA looks for clues made up with accessed documents almost consecutively. Figure 1 shows clues found by the CEA in a sequence of accessed pages by a user. The horizontal axis represents the time. In other word, each column corresponds to a single document. A square in a column means that the document belongs to a clue. Squares representing the documents that belong to the same clue are framed in a rectangle. The wordset attribute of a clue is displayed in the lower-right part of the window (Clue Wordset Att.).


VISS : Vizualization of Information Seeking Strategies - concept_plat132.227.204.77_soften_DYNAMIC_WINDOW_2_final
File Edit Vizualization Clues Interactions Help
Interactions of the clue
Wordset attribute of the clue
Date of the interaction
URL of the interaction
Shift Control Panel
Set/Save
Delete
Zoom

Thu Jan 09 17:38:34 CET 2003
http://www.worldnews.com/
Type: -1
Clue Wordset Att.
work, special, january, art,

Fig 1. Example of clues found by the CEA

We have noticed that the issue of extracting clues can be compared to the issue of extracting of lexical chains in a document [13]. A lexical-chain in is a sequence of terms in a text that are semantically related. The generic algorithm for building lexical chains is the following [13]:

- Begin with no chains
- For each word in the text
  - For each sense of the word
    - Among present chains, find the sense that is closest, and link the new word to this chain. Remove all other senses of the new word and the linked word.
    - Or, if no sense is close enough, then start a new chain with all senses of the new word.

Lexical chains have been used for text-segmentation [14]. Lexical chains serve to identify sentences that correspond to paragraph boundaries. Thus, they are represented by a sequence of terms and the set of sentences containing these terms. Accordingly, clues and lexical chains are both defined by terms and the structure which contain them. However, a lexical chain starts with one word and is updated when a semantically related term occurs in the remaining of the text whereas a clue starts with a given set of terms and is updated when the user accesses a new document containing the same words (more or less consecutively, see [4]).

In order to detect the user shifts, clues are removed when it becomes impossible that a new document belongs to an existing clue because of a temporal criterion. Then, the shifts are identified by the accessed documents that make the clue set be empty.

In the next section we present the result of an evaluation with real-users in order to compare the two previous approaches.

5. EVALUATION

5.1. Experimentation

In this section we present the details of a preliminary experimentation intended to compare the efficiency of the two previous methods to detect the users' shifts. The survey involved ten web-skilled participants. The number of participants may be not enough to generalize the results but it provides a framework for studying the tuning of the algorithms and comparing their strengths and weaknesses.

The participants were provided a questionnaire with 17 information seeking problems. They were allowed to answer the questions in whatever order they wanted but they could not perform parallel searches.

In the sequel, real shifts denote the times when the users end up with a question and start a new one. Questions in the questionnaire are designed so that the user has to perform different kind of information-seeking strategies (Table 1).

Q1	How many points did Michael Jordan score during his last match?
Q2	Find four movies that are played in cinemas close to your house tonight.
Q3	What is the ranking of the top 5 leaders on the US market of car sells?
Q4	What are the special exhibitions in the Modern Museum of Art (US, NY)?
Q5	What is the news today in India?
Q6	What is the price of a subway ticket in London?
Q7	How will be the weather be in Acapulco (Mexico)?
Q8	How much does a HEWLETT PACKARD Toner - C7115X - Black cost?
Q9	What is the city which has the longest name in the world?
Q10	What is the ethomilogia of rhabdomancy?
Q11	What is the ranking of the top 5 leaders on the EU market of computer sells.
Q12	What is the third highest mountain in the World?
Q13	Is Patras (Greece) larger than Pleven (Bulgaria)?
Q14	How many goals did Pelé scored in his career?
Q15	How many grand slams and WTA tournaments Carlos Moya won in his career?
Q16	In Greece, the two biggest cities are Athens and Thessalonica. What is the third one in Greece?
Q17	You want to find the lyrics of a George Harrisson's song. Yet you don't remember the name of the album and you just know that it is the last one he recorded before he died and it was the last song of the album. Find it.

Table 1. Questionnaire

After the experimentation, pages accessed by the users were downloaded; HTML tags and scripts were removed. We also used stopword list and stemmed the remaining terms with the Porter's stemming algorithm [15].

5.1. Evaluation

In this section the evaluation measures are presented. They are defined with respect to the possible shifts. Let us call t_f and t_l respectively the times of the real shifts corresponding to the beginning and the end of a question. To evaluate the predictions, three cases have to be considered:

- Only one detected shift is detected at time t with $t_f < t < t_l$. It implies that the algorithm has discovered that the user has started a new search activity. Then we call this predicted shift a *relevant shift*.
- Two or more possible shifts are detected at times between t_f and t_l. It means that the algorithm has detected more search activities than what really occurred. Then, the first predicted shift is a relevant shift and the followings are called *duplicated shifts*.

No real shift is detected. It means that the algorithm hasn't been able to detect that the user has started a new search activity. Then we say that the search activity is *missed*.

In our experimentation there are 17 questions and thus 17 real shifts to detect for each user.

Let *P*, *R*, *D* be the numbers of predicted, relevant and duplicated shifts respectively and M be the number of missed search activities, then:

$$P = R + D$$
$$M = 17 - R$$

It could also be interesting to consider the previous definitions within the classical decisional analysis framework. The predicted shifts are the predictions. The number of true-positives is the number relevant shifts. The number of false-negatives is the number of missed search activities. The number of false-positives is the number of duplicated shifts. The number of true-negatives is 0 as predicted shifts are either duplicated or relevant.

In decisional analysis, sensitivity and specificity are two classical criteria to compare and evaluate diagnosis algorithms. Sensitivity is the number of true-positives divided by all the real shifts. The better the sensitivity of the test is, the fewer the false-negatives are. The specificity is the number of true-negatives divided by the number of interactions that are neither a real shift nor a duplicated one. The better the specificity of the test is, the fewer the false positives are.

5.1. Results and Discussion

In average, it took about 58 minutes to each participant to answer the questionnaire. During this time, they performed an average of 192 interactions. Thus, the average number of interactions to answer a question was 11.3 though two questions required about 20 interactions each. Real shifts were manually annotated.

The two algorithms were then compared. We used the Choi's implementation of TextTiling, JTextTile[1]. We set *w*=200 because this value corresponds to the mean document size[2]. The algorithm was tested with different values for *s* (*s*=40,60,80,..,i,i+20,..,200).

For the CEA, we used the same tuning as the one described in [4]. We used a parameter *N* was chosen: It means that the CEA will use only the last *N* accessed documents to see if they have

[1] http://www.cs.man.ac.uk/~mary/choif/software.html

[2] In his experiments, Kaufmann [12] also sets w=200.

common points with the content of the current document. Furthermore, we removed all the clues having a number of words and interactions lower than 4.

Figure 2 gives the mean percentages of *R, D* and *M* with respect to the number of questions when the parameters *N* and *s* of the CEA and TextTiling vary respectively. The percentage of relevant shifts and missed search activities are computed with respect to the number of real shifts (17) while the percentage of duplicated shifts is computed with respect to the number of predicted shifts (*R*+*D*).

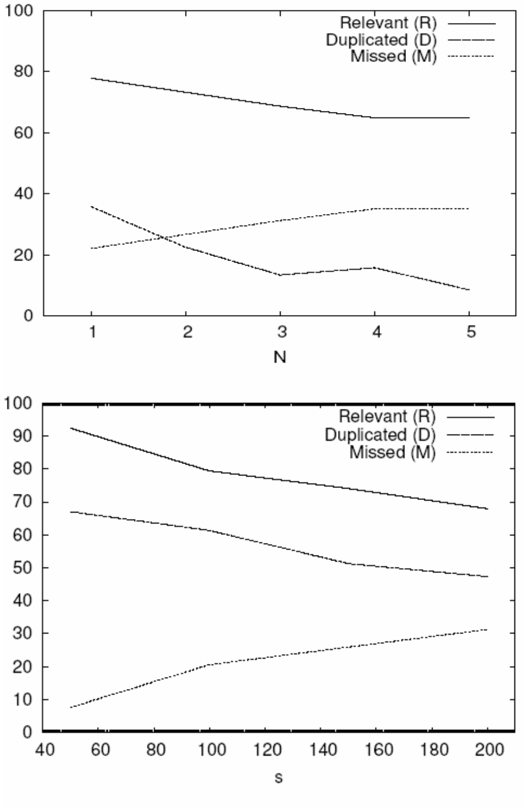
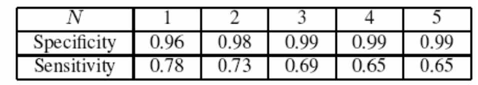


Fig 2. Performance comparison of the CEA (up) with TextTiling (down)

For the CEA, at *N* increment, the number of relevant shifts and duplicated shifts decreases, and accordingly the number of missed search activities increases. The same observation holds for TextTiling at *s* increment. Note that although the number of relevant shifts is in this case higher than with the CEA, the number of duplicated shifts (false detection) is also higher.

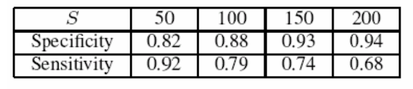
It is a drawback of TextTiling to require two blocks of words (on the left and on right of the current position) to detect a shift. On the other hand, the clue-based approach is also dependent on the value of *N* and a real shift may be detected up to *N* interactions too late.

The lexical cohesion-based approach is more sensible than the clue-based approach. On the other hand, the latter has a higher specifity than the former. Tables 2 and 3 summarize the sensitivity and sensibility of the two approaches when *N* and *s* respectively vary:



N	1	2	3	4	5
Specificity	0.96	0.98	0.99	0.99	0.99
Sensitivity	0.78	0.73	0.69	0.65	0.65

Tab 2. Sensitivity and specificity of CEA



S	50	100	150	200
Specificity	0.82	0.88	0.93	0.94
Sensitivity	0.92	0.79	0.74	0.68

Tab 3. Sensitivity and specificity of TextTiling

5. CONCLUSIONS AND FUTURE WORKS

This article has presented a novel approach to unobtrusively detect the user's shifts from the accessed contents. It is based on the assumption that the issue of detecting shifts of focus is isomorphic to the issue of detecting paragraph boundaries in a document made up with the previously accessed contents. The paper also showed that the shift detection algorithm proposed in [4] can be seen as a kind of text-segmentation algorithm based on lexical chains. A preliminary evaluation with real-users shows that both approaches have a high prediction rate. The proposed approach often comes up with more false-positive (higher sensitivity) and less true-negative (smaller specificity) than the approach based on the CEA.

Though the users where allowed to answer the questionnaire in whatever order, the questions were related to different topics and answered in a row. Future work will tackle to the task of detecting user shifts when they are performing parallel searches in different browser windows and when the information needs evolve but remain related on the same topic.

REFERENCES

[1] Claypool, M., Le, P., Wased, M., Brown, D. (2001) “Implicit interest indicators”, Proceedings of the ACM Conference on Intelligent User Interfaces, pp33-40.

[2] Hijikata, Y. (2004) "Implicit User Profiling for On Demand Relevance Feedback", In Proceedings of the ACM Intelligent User Interfaces, pp98-205.

[3] Lieberman, H. (1995) "Letizia: An Agent That Assists Web Browsing", In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp924-929.

[4] Delort, J.-Y. (2004) “Seeking for Clues About Users' Information Needs in Their Navigations”, Proceedings of IADIS International Conference WWW/Internet.

[5] Billsus, D., Pazzani, M. J. (1999) "A Hybrid User Model for News Classification", In Proceedings of the Seventh International Conference on User Modeling, pp99-108.

[6] He, D., Goker, A. (2000) "Detecting Session Boundaries from Web User Logs", In Proceedings of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research.

[7] Cooley, R., Mobasher, B., Srivastava, J. (1999) "Data Preparation for Mining World Wide Web Browsing Patterns", Knowledge and Information Systems.

[8] Delort, J.-Y., Bouchon-Meunier, B. (2002) "Link recommender systems: the suggestion by cumulative evidence approach" In Proceedings of STAIRS.

[9] Lau, T., Horvitz, E. (1998) "Patterns of Search: Analyzing and Modeling Web Query Refinement" In Proceedings of the 7th International Conference on User Modeling.

[10] Morris, J., Hirst, G. (1991) "Lexical cohesion computed by thesaural relations as an indicator of the structure of text", Computational Linguistics, Vol. 17, No. 1.

[11] Hearst, M. (1994) "Multi-paragraph segmentation of expository text", In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp9-16.

[12] Kaufmann, S. (1999) "Cohesion and collocation: Using context vectors in text segmentation", In Proceedings of the 37th Annual Meeting of the Association of for Computational Linguistics", pp591-595.

[13] Hirst, G., Budanitsky, A. (2001) "Lexical chains and Semantic distances", In Eurolan-2001.

[14] Stokes, N., Carthy, J. & Smeaton, A. F. (2002), “Segmenting Broadcast News Streams using Lexical Chaining”, In Proceedings of STAIRS, pp. 145-154

[15] Porter, M.F. (1980) “An algorithm for suffix stripping”, *Program*, Vol. 14, No. 3, pp130-137.