CEA: A Content-Based Approach of Shift of Focus Detection in Hypermedia Navigation

Jean-Yves Delort, Bernadette Bouchon-Meunier, Maria Rifqi

A Content-Based Approach of Shift of Focus Detection in Hypermedia Navigation

J.-Y. Delort
LIP6 - UPMC
8 rue du capitaine Scott
75015 Paris, France
jean-yves.delort@lip6.fr

B. Bouchon-Meunier
LIP6 - UPMC
8 rue du capitaine Scott
75015 Paris, France
bernadette.bouchon-meunier@lip6.fr

M. Rifqi
LIP6 - UMPC
8 rue du capitaine Scott
75015 Paris, France
maria.rifqi@lip6.fr

ABSTRACT

Adaptive Navigation systems (ANS) are intended to support user navigation in a hypermedia system. In the context of the Web, ANS suffer from the difficulty of finding relevant criteria to represent the users' current information needs and their frequent shifts of focus. This paper addresses the issue of user's information needs modeling. The proposed approach is to find relevant clues about the user's current interests from the content of the accessed pages. First, the concept of clue about user's information needs is formalized. Secondly, a clue extraction algorithm in a stream of accessed pages is presented. Then, *VISS*, a visualization tool intended to display user's different search activities with respect to clues found in a stream of accessed pages is put forward. Finally, this paper discusses the result of a Web navigation survey intended to characterize user's shifts of focus.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*User issues*; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*navigation, theory, user issues*

General Terms

Algorithms, Experimentation

Keywords

Hypermedia navigation, information needs, shifts of focus

1. INTRODUCTION

Adaptive Navigation systems (ANS) are intended to support user navigation in a hypermedia system. Applications range from recommender systems to tour generation through online learning[1]. ANS are able to adapt to users' behaviors, *i.e.* their past and current preferences, information needs and navigation styles. That is why ANS can be seen as activity monitoring applications: they use a stream of user's browser interactions and they have to adapt to the changes they detect. On the Internet, ANS suffer from:

- the difficulty of understanding users' current information needs,
- the frequent users' shifts of focus.

Lately, context-based ANS using the content of the accessed documents by the users have been proposed [9, 10, 7, 15]. Each of these works concluded that relevant information about the user's current information needs could be extracted from the content of the accessed documents. In many information-seeking models, interactions are indeed directly connected to users' information needs. According to Marchionini [14], the characteristics of the users' current information needs are used to specify and guide the search process and thus the interactions. But conversely, as demonstrated in the Bates' evolve/berrypicking model [1], users' interactions are also likely to make their current information needs evolve: as a consequence of their viewing the intermediary result sets, features they have in mind of their current information needs may be changed, removed or added. To sum up, the content of the documents accessed during a search activity is likely to:

- contain clues about the user's current information needs,
- trigger user shifts of focus or strategies.

This paper makes the following contributions. First, the concept of clues is formalized in the context of relevant pieces of information taken from the content of the accessed documents during a search activity. Secondly, a complete incremental algorithm designed to extract the clues from a stream of interactions is described. Third, *VISS*, a tool for visualizing these clues is put forward. Finally, the results of an experimentation intended to study the user's shifts of focus on the basis of these clues is presented.

2. RELATED WORKS

The concept of user session is often used to refer to relations between the users' information needs and their interactions: it represents a set of accessed pages related to the same search activity. User session boundaries correspond to user shifts.

[1] see for instance [2] for a review of the different existing applications.

Various time-based detection processes of session boundaries based on their interactions have been proposed [11, 6]. For instance, in [8] user sessions are extracted with respect to a maximal time span between consecutively accessed pages. Catledge and Pitkow [5] discovered a relation between the length of successively accessed pages and their frequencies. They derived from this relation a characterization of the user's information-seeking behavior. Canter et al. [4] have formally identified five types of search activities in a hypermedia system: wandering, browsing, scanning, exploring and searching. For instance, scanning refers to covering a large area without depth. In their model, a search activity corresponds to a specific combination of low-level browsing patterns (spikiness, ringiness, loopiness and pathiness) that can be taken from the path followed by the user. Mullier [16] developed a fuzzy rule based system to classify search activities in the Canter's classification.

Users' information needs have also been studied with respect to their search query to web search engines [3, 13, 12] or their behavior in Web directory [3]. For instance, analyzing web search queries recorded by a popular search service, Lau and Horvitz [13] identified seven classes of shifts, and proposed a classifier to predict the next user shift with respect to his previous queries.

3. BACKGROUND

General definitions as well as the concept of clue are formalized in this section.

3.1 Definitions

Navigation interactions are a type of browser interactions the user does in order to see a different page from the one displayed in the current browser window. Examples of navigation interactions are: clicking on the back button, submitting a form or clicking on a link. Examples of non-navigation interactions are: adding a bookmark or searching a pattern of words in the page. In the sequel an *interaction* refers to a navigation interaction. An interaction i is characterized by the pair (t, d) where t is the time when the user did the interaction and d is the *associated document* consequence of the interaction i. d is to be considered as a set of words. Numeric values are possible thanks to a representation of documents with fuzzy sets.

An *interaction stream* S is a strictly ordered set of interactions by a given user with respect to the time attribute. The size of an interaction stream is the number of interactions it contains. The set $P(S)$ will refer to the powerset of S.

3.2 Clues

Recent researches in different ANS fields (link generation [9], recommender systems [15], search in context [10] and prefetching [7]) have used the content of the current or previously accessed documents to get relevant clues about the user's current information needs.

Marchionini [14] defines two main styles of information-seeking behavior: during a searching activity (resp. a browsing activity) the user has in mind specific features (resp. general features) or characteristics of the objects that will be used to satisfy his information needs. He stresses that these features are used to guide and specify the search activity.

We argue that during a search activity, the content of the accessed documents often contains relevant terms with respect to the user's information needs. Assume that the user's goal is to find when the Paris Eiffel Tower was built. Then the previous hypothesis suggests that the content of accessed documents during the search activity should frequently contain terms such as "Eiffel Tower", "Paris" or "France". Furthermore, additional terms discovered during the search activity and related to the information needs are also likely to appear frequently, for instance "International Exposition"[2].

DEFINITION 1 (clue). *Let T be the interaction stream.A* clue in the stream S *is a pair (I, W) such that:*

1. $I \in P(T) \setminus \emptyset$, *where $P(T)$ refers to the powerset of T.*
2. $W = \bigcap_{i_k \in I} d_k$ *where d_k is the associated document of k^{th} interaction i_k in the stream T.*
3. $W \neq \emptyset$

4. THE CLUE ENUMERATION ISSUE

The clue enumeration issue refers to the problem of enumerating all the possible clues in an interaction stream. A simple enumeration to find all the pairs can contain up to 2^n clues for a stream of size n. However, many of the clues found are likely to be irrelevant because not corresponding to a user's realistic search activity. A clue $c = (I, W)$ will be considered as an irrelevant clue if I does not belong to a realistic search activity. Accordingly, the key issue in the clue enumeration problem is to choose conditions for a subset of interaction I to be a part of one realistic search activity. Formally, given an interaction stream S, the condition for a subset of interactions to be part of one realistic search activity can be formalized by a boolean function F on the powerset of S. Thus, given F and S, the set of *relevant clues* is the set of clues R such that:

$$R = \{c \ : \ c \text{ is a clue in } S \text{ and } F(c) = 1\}$$

4.1 Neighborhood conditions

Let S be an interaction stream. The likelihood for a subset of interactions to belong to the same search activity is modelized through a neighborhood condition. A *neighborhood condition* has the property that, if true for a set I then it is also true for any subset $J \subseteq I$. Accordingly it can be formalized by a boolean measure m defined on the powerset of S such that:

$$m(A) = 1 \implies \forall B \subseteq A,\ m(B) = 1 \qquad (1)$$

This measure will be referred to as a *neighborhood measure*. Examples of neighborhood measures are:

1. Suppose that the neighborhood condition implies that the time span between two consecutive interactions in a subset of interaction never exceeds a fixed threshold of N_1 seconds. If $J = \{j_1, .., j_k\} \subseteq S$ is a subset of interaction of size k (with $k \geq 1$), let σ be the permutation of $(1, .., k)$ such that $J = \{j_{\sigma(1)}, ..j_{\sigma(k)}\}$ is an interaction stream. Then $m(J) = 1$ *iff*:

$$\forall j,\ 1 \leq j < k \quad t_{\sigma(j+1)} - t_{\sigma(j)} \leq N_1$$

2. Suppose that the neighborhood condition states that the time span between the first and the last interactions of I never exceeds a fixed threshold of N_2 seconds. If $J = \{j_1, .., j_k\} \subseteq$

[2]The Eiffel Tower was built as a focal point for the Paris International Exposition (public exhibition) of 1889.

S is a subset of interaction of size k (with $k \geq 1$), let σ be the permutation of $(1, .., k)$ such that $J = \{j_{\sigma(1)}, ..j_{\sigma(k)}\}$ is an interaction stream. Then $m(J) = 1$ *iff*:

$$t_{\sigma(k)} - t_{\sigma(1)} \leq N_2$$

3. Suppose that instead of using a time span we take into account a maximal number of intermediary interactions N_3. As interactions are strictly ordered with respect to their time attribute then a ranking function r can be introduced: $r(i_l) = \sum_{j=1..n} 1_{t_j < t_l}(i_j)$ where t_k is the time attribute of interaction i_k (r gives the rank of an interaction in S). Let σ be the permutation of $(1, .., k)$ (with $k \geq 2$) such that $J = \{j_{\sigma(1)}, ..j_{\sigma(k)}\}$ is an interaction stream. Then $m(J) = 1$ *iff*:

$$\forall j,\ 1 \leq j < k \quad r(i_{\sigma(j+1)}) - r(i_{\sigma(j)}) \leq N_3$$

4.2 Ending conditions

Let $S = \{i_1, .., i_n\}$ be an interaction stream and $1 \leq k \leq n$. Let denote by C_k the set of elements of $P(\{i_1, .., i_k\})$ which can still be part of a future search activity after interaction i_k. An *ending condition* is intended to state whether a set of interactions belongs to C_k or not. An ending condition has two fundamental properties:

1. At time k, interaction $\{i_k\}$ belongs to C_k,
2. $\forall I \in C_{k+1},\ I \setminus \{i_{k+1}\} \in C_k$. In other words, the set of future possible search activities only depends on the set of future possible search activities before the last interaction occurred.

Ending conditions can be formalized by an *ending function* which is a multivariate function $V : \mathbb{N}^* \longrightarrow P(S)$ such that:

1. $\forall k,\ \{i_k\} \in V(k)$
2. $\forall I \in V(k+1), I \setminus \{i_{k+1}\} \in V(k)$

A very simple example of ending function is to consider that the only subset of interactions containing an interaction occurred after the last N interactions (with $N \geq 1$) are likely to be in a current or future search activity. Let $S = \{i_1, .., i_n\}$ be an interaction stream and $1 \leq k \leq n$, $K = max(0, k - N)$ then:

$$V(k) = \{X \in P(S) \ :\ \{i_K, i_{K+1}.., i_k\} \cap X \neq \emptyset\}$$

5. CLUE EXTRACTION ALGORITHM

In this section, the *Clue Extraction Algorithm* (CEA), an incremental algorithm for finding clues in an interaction stream S is presented. The condition measure F for a subset of interactions to be part of realistic search activity is supposed to be given by a neighborhood condition m, thus $F(I) = m(I)$. To keep the solution set size reasonable, the proposed algorithm extracts clues having maximal number of interactions: a clue $c = (I, W)$ in a stream S will be retrieved if I is the largest set of interactions in the stream that possesses all the words of W with regard to F. The proposed algorithm is based on the incremental extraction of *active clues* which are considered to be a good caracterization of relevant clues in the stream S.

DEFINITION 2 (active clue). *Let $S = \{i_1, .., i_n\}$ be an interaction stream, m be a neighborhood measure defined on $P(S)$, V be an ending function and $1 \leq k \leq n$. An* active clue *at time k is a pair $c = (I, W)$ where:*

1. $I \in P(\{i_1, .., i_k\}) \setminus \emptyset$
2. $W = \bigcap_{i_k \in I} d_l$ *where d_l the associated document of interaction i_l*
3. $W \neq \emptyset$
4. $I \in V(k) \cap \{J \in P(\{i_1, .., i_k\})$ *such as* $m(J) = 1\}$

Using active clues has the two interesting advantages:

1. The set of active clues A_{k+1} after the interaction i_{k+1} depends only on A_k. Thus, its elements can be computed recursively. Indeed, let $(I, W) \in A_{k+1}$ with $I \neq \{i_{k+1}\}$, $J = I \setminus \{i_{k+1}\}$ and $W' = \cap_{i_l \in J} d_l$ where d_l is associated document of interaction i_l. Clearly $W' \neq \emptyset$. According to equation (1), $m(J) = 1$ and according to property 2 of ending conditions, $J \in V(k)$. Accordingly, the clue (J, W') is an element of A_k and c can be easily computed from it.
2. Let $c = (I, W)$ be an active clue in a stream S at time k. Then if $\big((I \cup \{i_{k+1}\}, W) \notin A_{k+1}\big)$ or if $(c \notin A_{k+1})$, there will no be any later interaction i_p at time p (with $p > k + 1$) such as $(I \cup \{i_p\}, W) \in A_p$ or $(c \notin A_p)$. This property can easily be proved using the definitions of ending functions and neighborhood functions.

The following algorithm extracts the set C of possible clues of the clue enumeration issue from an interaction stream $S = \{i_1, .., i_n\}$ of size n. The neighborhood measure and the ending function are denoted by m and V respectively.

```
CLUE EXTRACTION ALGORITHM(S)
    A_0 ← ∅
    C← ∅
    for k ← 0..(n − 1) do
        Take interaction i_{k+1} ∈ S
        A_{k+1} ← ∅
        for all (I_p, W_p) ∈ A_k
            (I, W) ← ({i_{k+1}} ∪ I_p, d_{k+1} ∩ W_p)
            if W ≠ ∅ then
                if W ⊂ W_p then
                    if I_p ∈ V(k + 1) then
                        A_{k+1} ← UPDATE(A_{k+1}, (I_p, W_p))
                    endif
                    if m(I) ≠ 0 and I ∈ V(k + 1) then
                        A_{k+1} ← UPDATE(A_{k+1}, (I, W))
                    endif
                else
                    C ← C ∪ (I_p, W_p)
                endif
            else
                if I_p ∈ V(k + 1) then
                    A_{k+1} ← UPDATE(A_{k+1}, (I_p, W_p))
                else
                    C ← C ∪ (I_p, W_p)
                endif
            endif
        endfor
        if m({i_{k+1}}) = 1 then
            A_{k+1} ← A_{k+1} ∪ (i_{k+1}, d_{k+1})
        endif
    endfor
    C ← C ∪ A_n
    return C
```

The worst-case complexity of one iteration of this algorithm is $O(N_k^2 \times |d_{k+1}|)$ where N_k is the number of clues in A_k and $|d_{k+1}|$ refers to the number of words in the document associated to interaction i_{k+1}. Note that the size of N_k depends only on the neighborhood condition and the ending condition. The clue extraction algorithm uses an update function that adds the pair (I, W) to the set A_k if and only if A_k does not already contain a pair (I', W') with $I \subseteq I'$ and $W \subseteq W'$. If the pair to add is larger than an existing pair, then the previous pair is replaced by the new one. Thus elements of A_k are clues. This update function is given in the following algorithm:

```
UPDATE (A_k, (I, W))
    for all (I_p, W_p) ∈ A_k
        if I_p ⊆ I and W_p ⊆ W then
            A_k ← (A_k \ (I_p, W_p)) ∪ (I, W)
            return A_k
        endif
        if I ⊆ I_p and W ⊆ W_p then
            return A_k
        endif
    endfor
    return A_k
```

Let us see an example. Assume that $S = \{i_1, .., i_6\}$ is an interaction stream. The neighborhood function states that the maximal number of intermediary interactions is at most 1 and the ending condition states that the only previous interaction is likely to be a part of a future clue. Each associated document to interactions of S is described by a subset of words of $\{a, b, c, d, e\}$ such as:

	a	b	c	d	e
i_1	x	x	x		x
i_2	x		x		
i_3	x	x		x	
i_4				x	x
i_5			x		
i_6	x	x		x	x

In this array an "x" means that the document in column contains the word in row. Clues of the solution set given by the CEA are ranked with respect to the time attribute of their first interaction in the stream. Then, the clues can be represented in the following array:

	i_1	i_2	i_3	i_4	i_5	i_6
$\{a, b, c, e\}$	x					
$\{a, c\}$		x				
$\{a, b, d\}$			x			
$\{d, e\}$				x		
$\{c\}$					x	
$\{a, b, d, e\}$						x
$\{a\}$	x	x	x			
$\{a, b\}$	x	o	x			
$\{c\}$	x	x				
$\{d\}$			x	x	o	x
$\{d, e\}$				x	o	x

In this array, an "x" means that the associated document of the interaction in column contains the set of words in row and an "o" means that the associated document of the interaction in column does not contain the set of words in row but is likely to belong to the same realistic search activity. The first part of the array displays the single-element clues and the second part contains the clues which size of their interaction set attributes is larger than one. This second kind of clues show the relationships between the interactions.

6. VISS

Visualizing the clues is an efficient way to discover characteristics of user's information-seeking behavior. For instance, the presence of long-size clues over a time span in a stream implies that the user is likely to have performed a searching activity (this will be demonstrated in the result section). Conversely, small-size clues imply that the user is likely to have performed browsing activity such as reading an online newspaper. In the previous array two clusters of clues can be easily discovered. It is then likely that the user has had two distinct search activities.

VISS (for Visualization of Information Seeking Strategies) is a visualization tool that displays the solution set of the clue enumeration issue for a given stream. VISS represents the solution in the same way as in the previous example (see figure 1). Each clue is associated with a unique color and is framed in a rectangle that shows all the interactions that are likely to belong to the same realistic search activity. A square in a rectangle means that the interaction in column contains the set of words in row. Word sets (associated documents in interactions) are displayed at the bottom of the window when a clue is selected. VISS has the following functionalities: searching the clues that contain a given set of words, zooming, highlighting more or less specific clues and annotating interactions and clues.

VISS can call clustering algorithms on a selected substream of interactions through the Weka package[3] and then display their results.

7. EVALUATION

This section summarize the results of a survey intended to assess the ability of clues to detect user's shift of focus. The first subsection outlines the survey intents. In the second subsection results are presented and discussed.

7.1 Survey

A survey was conducted to assess the interest of clues to find relevant features of users' current information needs. This section presents the results related to the issue of detecting users' shifts of focus. Results must be considered as preliminary because the number of survey participants is low (until now 10 participants). A questionnaire with 17 information seeking problems was given to the participants. Each person was asked to find on the Web documents containing answers to the given problems. Users' interaction streams were collected thanks to a modified version of the Mozilla browser they used to answer the questions[4].

Questions in the questionnaire were designed so that they involved the users in different information-seeking strategies:

1. Some questions were harder, if not impossible, to answer just using a search engine. The interest of these questions is to see if relevant clues can be extracted from documents which have not been accessed by means of user queries. For instance:

[3]Weka is a collection of machine learning algorithms written in Java, http://www.cs.waikato.ac.nz/ml/weka/.

[4]Mozilla is a GNU Web browser, available at http://www.mozilla.org.

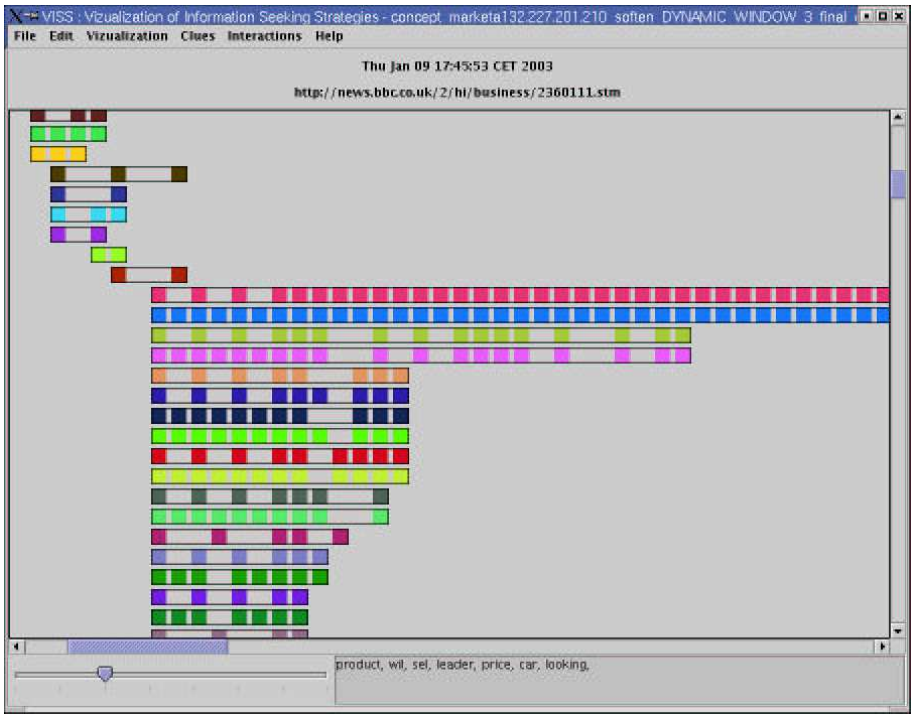

Figure 1: Clue visualization with VISS

"What is the exhibitions schedule of the Museum of Modern Art, NY?"

2. Some questions required to see at least more than one page to be answered. Such questions were intended to see if relevant clues can be extracted with search strategies combining smaller search strategies. For instance, "Which one of these two cities, Patras (Greece) and Vidin (Bulgaria) has the larger population?".

3. Some questions were really hard to answer, so that few participants could answer them. The interest of these questions was to require really complex search strategies when the user try to address the question from different points of view. For instance, "What is the ranking of the top five personal computer resellers in the EU market?".

Note that participants were allowed to answer the questions in whatever order. Each associated document was then stemmed with the Porter's stemming algorithm [17]. A stopword list was also used to filter irrelevant words, such as "these", "those"... The clue extraction algorithm was then applied on each user interaction stream. The tested ending conditions state that the only subsets of interactions containing an interaction occurred after the last N interactions are likely to be in a current or future search activity (associated ending measures are formalized in a previous example). The tested neighborhood conditions state that interaction within a clue cannot be spaced of more than a maximal number of M interactions (associated neighborhood measures are formalized in the third example of neighborhood condition subsection). In this paper results are given for $N = M$ and N varies from 1 to 5.

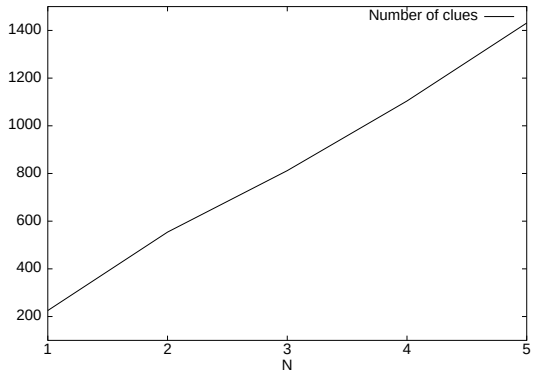

Figure 2: Clues set size increases quickly

7.2 Results and discussion

In average, it took about 58 minutes to the participants to answer the questionnaire. During this time, they performed 192 interactions in average. Thus, the average number of interactions to answer a question was 11.3 though two questions required about 20 interactions each. The share of search engines result pages, cached documents and homepages over the total number of accessed documents represents 40%.

Figure 2 displays the average number of extracted clues from the participants' interaction streams when N varies. The solution set size tends to increase quickly every N increment.

Let F be a condition measure for a subset of interactions to be part of a realistic search activity. Each interaction can be characterized

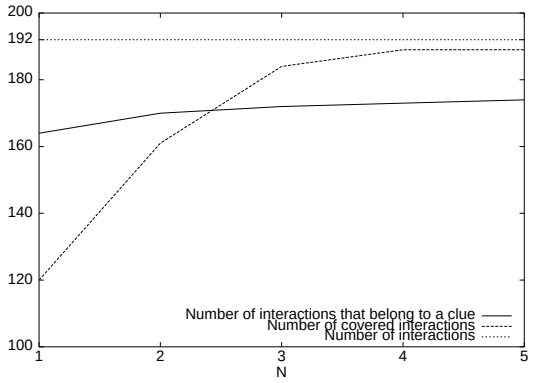

Figure 3: Relations betweens interactions and clues

with respect to the solution set C of the clue enumeration issue. Given an interaction i_k occurring at time k, i_k is *covered* by a clue $(I, W) \in C$ if $F(I \cup \{i_k\}) = 1$. In other words, i_k is covered by a clue if it occurs while clues about search activities have previously been detected and are still running. An interaction i_k *belongs* to a clue $(I, W) \in C$ if $i_k \in I$. With VISS, interactions are represented in columns and thus, the set of clues covering an interaction corresponds to all the rectangle intersecting the associated column (see figure 1). In the previous example, interaction i_5 is covered by the clues $(\{d\}, \{i_3, i_4, i_6\})$ and $(\{d, e\}, \{i_4, i_6\})$, but does not belong to any of them. The set of clues which interaction set attributes contain a given interaction correspond to all the rectangles that have a colored square in the column associated to the interaction. For instance with the example in section 5: i_5 is covered by the clues $(\{i_4, i_6\}, \{d, e\})$ and $(\{i_4, i_5, i_6\}, \{d\})$ but only belongs to $(\{i_4, i_5, i_6\}, \{d\})$.

Figure 3 displays the number of interactions that belong to a clue and are covered by a clue when N varies. Though many new clues are discovered with an N increment, few interactions that did not belong to a clue of the previous clue enumeration issue solution set will eventually belong to one of the newly discovered clues. However the number of interactions that are covered by a clue increases: when $N = 5$ about 98.4% of interactions are covered while there are just 62.5% when $N = 1$.

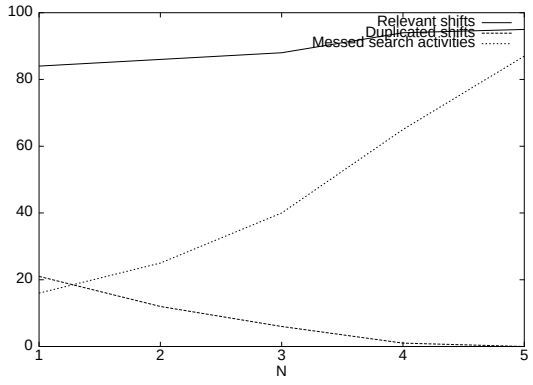

Figure 4: Shifts detection performance

A shift of focus corresponds to the situation where the participant has just finished answering a question and begins answering a new one. Thus, a pair of two consecutive shifts of focus corresponds to a real search activity boundaries. An expert identified and annotated in each participant's interaction stream the interactions corresponding to shifts of focus.

An interaction i_k can then be described with respect to three boolean attributes:

1. $hasPreviousActiveClue = 1$ *iff* the previous interaction to i_k is covered by a clue.
2. $hasActiveClue = 1$ *iff* i_k is covered by a clue.
3. $overlap = 1$ *iff* $A_k \cap A_{k+1} = \emptyset$ where A_k is the set of active concepts at time k.

Each interaction within an interaction stream was automatically tagged as a *possible shift* if :

$$hasActiveClue \wedge (\neg overlap \vee \neg hasPreviousActiveClue)$$

A possible shift is a *relevant shift* if it occurs during a search activity. A possible shift is a *duplicated shift* if a previous relevant clue was found within the same search activity boundaries. A search activity is *messed* if no one relevant shift was found within its boundaries. Given N, let P, R and M be the number of possible shifts, the number of relevant shifts and the number of missed search activities respectively, then[5] :

$$\begin{aligned} P &= R + D \\ M &= 17 - R \end{aligned}$$

Figure 4 gives the average percentages of P, R and M with respect to the number of questions when N varies. At N increment, the number of relevant shifts increases. As the number of covered interactions increases, the number of possible shifts decreases and thus, the number of messed search activities increases while the number of duplicated shifts decreases. Note that the average minimum number of interactions necessary to find a relevant shift after a user's shift of focus is about 1.7 interactions. Note that the best compromise between P, R and M corresponds to the case where $N = 1$.

Though the users where allowed to answer the questionnaire in whatever order, the questions were related to different topic. This explain for a part, the good performance the system to detect shifts of focus. In real estate the user shifts could be harder to detect if the user information needs were to change but still be related to the same topics. This would require more complex features (such as originality) to represent the interactions that would be used to characterize the possible shifts.

8. CONCLUSIONS

This work addressed the issue of detecting relevant clues about users' current information needs during their navigation on the Web. The originality of the proposed approach lies in the use of the content of the accessed documents to extract these clues. An incremental algorithm intended to extract clues from a user interaction stream was presented. Preliminary results show good ability of this approach to detect user's shifts of focus.

[5]Let us recall that 17 is the total number of questions in the questionnaire.

Clues are an efficient way to modelize user information needs. They are a rich source of information about the user because they link his behavior (his interactions) with the content of the accessed documents. For instance, user's interactions could be characterized in term of originality comparing the word set attributes of recently discovered clues to word set attributes of prior clues. User models based on clues can be used to study complex user's behavior such as their information-seeking strategies. Clues can also be used by ANS that require to know the current state of the user search activity and his current information needs. Future works will focus on the relations between users' information-seeking strategies and the clues that can be extracted from their interaction streams.

9. REFERENCES

[1] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, Oct 1989.

[2] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11:87–110, 2001.

[3] F. Cacheda and A. Vina. Understanding how people use search engines: a statistical analysis for e-business. In *Proceeding of e-Business and e-Work Conference and Exhibition*, 2001.

[4] D. Canter, R. Rivers, and G. Storrs. Characterizing user navigation through complex data structures. *Behaviour and Information Technology*, 4(2):93–102, 1985.

[5] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, 1995.

[6] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[7] B. D. Davison. Predicting web actions from html content. In *Proceedings of the The Thirteenth ACM Conference on Hypertext and Hypermedia (HT'02)*, pages 159–168, College Park, MD, June 2002.

[8] J.-Y. Delort and B. Bouchon-Meunier. Link recommender systems: the suggestion by cumulative evidence approach. In *Proceedings of STAIRS, Lyon, France*, 2002.

[9] S. R. El-Beltagy, W. Hall, D. D. Roure, and L. Carr. Linking in context. In *Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 151–160. ACM Press, 2001.

[10] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: the concept revisited. In *Proceedings of the Tenth International World Wide Web Conference*, 2001.

[11] D. He and A. Goker. Detecting session boundaries from web user logs. In *Proceedings of of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*, 2000.

[12] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: A study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.

[13] T. Lau and E. Horvitz. Patterns of search: Analyzing and modeling web query refinement. In *Proceedings of the Seventh International Conference on User Modeling*, 1998.

[14] G. Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, 1995.

[15] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu. Combining web usage and content mining for more effective personalization. In *Proceedings of the International Conference on E-Commerce and Web Technologies*, 2000.

[16] D. Mullier. Examining how users interact with hypermedia using a neural network. In *Proceedings of ICAI-00, Las Vegas, H. R. Arabnia (ed)*, 2000.

[17] M. F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, July 1980.