Capturing Information Need by Learning User Context

Ayşe Göker
School of Computer and Mathematical Sciences
The Robert Gordon University
St. Andrews Street, Aberdeen, AB25 1HG, Scotland, U.K.
Email: asga@scms.rgu.ac.uk

Abstract

Learning techniques can be applied to help information retrieval systems adapt to users' specific needs. They can be used to learn from user searches to improve subsequent searches. This paper describes the approach taken to learn about particular users' contexts in order to improve document ranking produced by a probabilistic information retrieval system. The approach is based on the argument that there is a pattern in user queries in that they tend to remain within a particular context over online sessions. This context, if approximated, can improve system performance. While it is not uncommon to link the evidence from one query to the next within a particular online session, the approach here groups the evidence over different sessions. The paper concentrates on the user-oriented evaluation method used in order to determine whether or not the approach improved information retrieval system performance.

1 Introduction

The purpose of information retrieval is to address the information need of a user, at a particular point in time. The argument here is that each information need has an associated *context*. Additionally, often a number of information needs will have a common context. Hence, what may be learnt from meeting one need may be of use for further ones. The work here is concerned with the actual information need and context of individual users.

Users will tend to repeat searches or conduct a series of closely related searches over a period [Walker and Hancock-Beaulieu, 1991]. Although each search must be regarded as representing a different information need, they can all be assumed to have a common context. A document that is judged relevant to the need which prompted one search will not necessarily be relevant to the next need, but is relevant to the context within which the need falls.

Frequent users of an information retrieval system (IRS) should be able to benefit from their high use of the system. Research has shown that users tend to have two to three topics on which they focus their queries [Goker and McCluskey, 1991]. This supports the need for learning or approximating the user's context to help guide future searches of the user. Additionally, evidence shows that the majority of user queries contain less than three terms [Walker and Hancock-Beaulieu, 1991; Jansen *et al.*, 1998] which is usually insufficient to describe the user's need.

To make an IRS more adaptable, the approach here is to guide the user, using a context learning technique, to documents which answer the information need. Thus, documents retrieved in response to a user's query are reordered using information from the user's context. This is implemented by integrating a Context Learner (CL) within a probabilistic IRS (Okapi, [Robertson, 1997]). In the end, it is only the user who can assess whether the query has been satisfied within the context of the current problem. Different users can have the same query but the levels and and depths of knowledge and expectations they have differ. Therefore, a user-oriented evaluation was undertaken.

This paper reviews various user-oriented approaches for accessing or retrieving information. It then goes on to look at an alternative approach, referred to as the Context Learner (CL). The problem domain, the learning method and how it was utilised is then described. This is followed by the results of experiments and the method of evaluation. Finally, findings are summarised and future work is discussed.

2 Related Work

Related work generally revolves around learning user profiles, meeting user defined goals, combining evidence from groups of users, and looking at recurrent patterns of document choices.

User profiles tend to represent interests of users over a long-term and typically are clear descriptions of certain kinds of documents. They usually focus on the topic of the query and are used to filter streams of incoming information. User profiles need not always be associated with a particular user, they can exist independently. Bloedorn et al. [1996] worked on automatically constructing user profiles by employing a generalisation hierarchy based on a thesaurus. These profiles are based on identifying a subject in the hierarchy which relates to the topic of the information need of a user. The software agent Syskill & Webert [Pazzani and Billsus, 1997] learns a profile from the user's ratings of Web pages. The profile is then used to suggest links which might be of interest to the user or help construct queries. It learns a separate profile for each topic of each user. The CUSTARD system [Edwards *et al.*, 1997] uses conceptual clustering to group similar documents together in order to treat them as concise profiles of users' information preferences.

Web Watcher [Armstrong *et al.*, 1997] is an information seeking assistant for the Web. The system suggests an appropriate link based on a link utility function derived from the text in the current page, the link in the page and the user goal. The user goal is a component based on explicit statements by the user (words entered by the user when filling in the form on the Web).

A case-based reasoning system Broadway [Jaczynski and Trousse, 1998] reuses past navigations of groups of users to recommend Web pages to visit next. However, the hypothesis is different to that behind the work described here. The CBR system is based on the hypothesis that if users access similar sequences of similar documents they may have similar browsing intent.

A different approach [Chalmers *et al.*, 1998] is to treat each URL (Universal Resource Locator) as a symbol and focus on recurrent patterns of symbol use that emerge from the context of activity. This approach uses statistical patterns of symbol recurrence in representing relevance and information need.

The work described in this paper differs to those above in several ways. Unlike user profiles, a user context is not about referring to the particular topic of a query but rather addressing the context in which it occurs. As regards capturing user goals, the work here focuses on the users' past use of the system rather than any initial description of intent for the current task. Previous online queries for a particular user are used specifically for that user and previous evidence from groups of users is not merged currently. Also, focus is not on the access patterns of documents by users in general but the sequence of queries and documents found to be relevant by a particular user. The conceptual framework of the work described in this paper is outlined in the next section.

3 The Problem Domain

The task is to approximate incrementally a particular user's context. The work can be viewed within the framework of the ASK model [Belkin *et al.*, 1982]. According to this model, a person with goals, intentions or a problem to resolve finds him/herself in a problematic situation. A characteristic of this situation is the Anomolous State of Knowledge (ASK) that the person tries to resolve. However, as the internal resources have been inadequate, the person has recourse to some external knowledge resource such as a a collection of text, organised and represented in some manner. A possible response to an ASK is an information need and this in turn can initiate a query. Context provides some of the knowledge needed to resolve the anomaly. The difficulty is that most users do not provide this context as part of the query and so the CL addresses this point.

The work involved the testing of the following two main hypotheses. Firstly, whether the notion of context exists or not and whether it is static or dynamic (i.e. a shift in context over time). Secondly, whether document ordering for a particular user based on his/her context is improved.

Effectively there are two sub-hypotheses associated with the first: Consecutive contexts are strongly related to each other (i.e. equal) in which case the context is stable, over time; Consecutive contexts are weakly related to each other, in which case there is a shift in the context. Likewise, the second hypothesis about document ranking could be described further. The strong hypothesis asserts that a CL produces a substantial improvement to document ranking; the weak hypothesis asserts that it produces a minor improvement.

Experiments were performed to see whether context learning over user sessions would be useful to users and provide improved performance. The experiments described in this work provide a framework for evaluating learning techniques for IRSs and search engines.

4 The Context Learning Method

This section describes the inputs to the context learner, the method used and the way in which it is utilised.

4.1 Data Source and Collection

Possible sources for learning in information retrieval and the ones used in this work are given below.

1. Features in documents or the documents themselves.
   One source of information is provided by documents and their contents. In this work, the document-identifications (code provided by the bibliographic database) and the stemmed words (terms) in the documents relevant to queries are used (so-called positive feedback).

2. Structure of subject area and domain knowledge as represented by a classification scheme or thesaurus.
   A knowledge base of the subject hierarchy could be useful in classification tasks. However, incorporating this into a CL would involve additional assumptions about user contexts and their relation to the knowledge base. Hence, it was not used within the scope of the work described in this paper.

3. User searches and user search behaviour.
   This source includes both the previous and current user online sessions. Details of these sessions are typically found in transaction logs. In the standard IRS (Okapi), logs contain all user keystrokes (e.g. for a query or a menu choice) and system responses.

This work extracts relevant details for the CL from these logs. However, it should be noted that the context, as described in this paper, is not a formal model of user search behaviour.

4. Background knowledge.
This refers to knowledge about the user or the problem/task. The user could provide this explicitly (e.g. biography or a description of the background to the problem) or the knowledge could be elicited. This could be a topic for future work but, initially, the aim was not to interrupt the flow of the user-system interaction by asking for lengthy descriptions or cross-checking of automatically elicited background knowledge.

The core elements available are documents, queries, terms, and users. Most of these elements can be reduced to terms. In this sense, the work here can be referred to as context term learning.

A good time for obtaining information from these various sources is during relevance feedback. If the context is to include a set of features, relevance feedback can contribute to their selection.

The IRS used in this project (Okapi) provides transaction logs of user online sessions. From these it is possible to extract various details relating to user queries. As the logs get updated after each query they can also be used for an online learning system.

The items below are used by the learner and of these, the first three can be found in the transaction logs:

- the stemmed query terms
- the documents judged relevant by the user with respect to a query
- the index terms from the relevant documents
- the previous set of context terms

4.2 The Learning Method

The learner consists of four modules A, B, C and D. Each module has a specific purpose and effectively relates to a stage in the CL. The algorithms are in modules A to D. There are a number of algorithms in each module and so a particular CL would involve the choice of particular algorithm from module A, another one from module B and so on.

The approach taken, initially, was to develop a CL which learns a context which *minimally covered* the set of documents judged to be relevant (by the user). The assumption behind mimimal coverage is that users' relevant documents, when covered minimally with a set of terms, will have the least amount of 'noise' or redundancies for representing the context. This does not mean that it is the best and most accurate way to represent context. Indeed, there is a question as to whether it will be sufficient to represent it.

Thus, the main purpose was to find the combination which contains the smallest number of terms to represent all the relevant documents. However, minimal coverage as a theoretical aim may be difficult to ensure in practice. Apart from efficiency, there are other reasons why this is difficult practically without introducing other criteria. For example, let us assume that all weights (as calculated by the probabilistic IRS) of terms t_1, t_2, t_3 are equal and that t_1 occurs in documents d_1, d_2, d_3, and let us also assume t_2 occurs in d_1, d_2 and t_3 occurs in d_3, d_4, d_5, d_6. It is not clear whether t_1 and t_3 together should cover the documents or t_2 and t_3. A possible solution is to favour terms with high frequencies in documents and/or those with high weights.

Other approaches for representing the context were also developed. These include directly using frequencies of terms in documents, the past frequencies of terms over iterative contexts, the performance of the term in the immediately preceding iteration and using threshold values for the number of terms in a set. These are addressed by the specific algorithms under each module. However, the overall learning method can be summarised as follows:

Module *A*: Forming the set of relevant documents R.
Module *B*: Choosing terms from the query and R.
Module *C*: Merging the terms acquired in module B with the previous context to create the new context. This is done in conjunction with module D.
Module *D*: Specifying the role of terms in previously acquired context.

Module *A*
    Choose R_{total} OR R_{latest}
    Either (remove duplicates) OR
    (do not remove duplicates)
where
R_{total} : *Set of relevant documents from all queries*
R_{latest} : *Set of relevant documents from the last query*

Module *B*
    Choose approximation method to context
    If applicable to algorithm chosen then
        apply threshold values or proportions for
        terms deriving from current iteration (only)
e.g. minimal coverage would be one of the algorithms under this module.

Module *C*
    Merge C_{n-1} with C_n based on
    (any threshold values for the whole context C_n) AND
    (criteria about previous terms in context
    - defined in module *D*)

Module *D*
    Choose the possible role of previously
    acquired terms in current context.

As an example of a particular CL, CL14 uses modules A^I, B^{IV}, C^{II} (with value 30) and D^{III}.

A^I: Use R_{total} with duplicates.
B^{IV}: Use minimal coverage.
C^{II}: Apply merging criteria but limit number of terms in resulting set to 30.

D^{III}: If the term was used previously then continue to use it in this iteration. If the term was not used in the previous context and has a low priority after module B, then do not use it in current iteration (also see Section 4.4 for description of used terms).

The data was gathered from online sessions but was later processed in batch mode. The procedure for performing learning in batch mode is as follows:

1. All online sessions, for a particular user, that fall within a specified time period are identified and their chronological order kept.
2. Each session is broken down into the queries that form it. Each query chunk includes the query and any relevance feedback that goes with it and the results of the query.
3. From these chronologically ordered chunks, the ones from which there was no positive relevance feedback are removed.
4. The remaining query chunks form a query data file each. These are in the form that the learner can accept as its input.

4.3 The Mode of Learning

Two versions, online and off-line, of the learning system were built. Initially, the system was implemented to work online (live mode). In this system, contexts were derived in real-time as the users performed their queries. However, an off-line (batch mode) system was also written in order to be able to reuse queries from previous sessions when testing different versions of the CL. The batch learner was the one used for the experiments.

In incorporating a learning technique into an IRS there are several issues that need to be addressed. Firstly, the method of deciding when to learn has to be established. There should be a delimiter of a query before the start of the next one so that any (positive) relevance feedback obtained from the user during that query is input to the CL. Delimiters can be performing a new query, editing the old one or exiting the system, for example. Secondly, a decision has to be made as to when to utilise the context learnt.

In both systems, the data comprised

- User session logs.
- Query files. These are the queries after the logs have been parsed and translated into the format ready for the CL.
- Context files and History files (record of previous contexts).
- Retrieved document details for queries after the CL has been put to use.
- User relevance judgements for documents, used in evaluation.
- Precision values (proportion of retrieved documents that are relevant) for the document lists.

4.4 Re-ordering Documents according to Context

Document reordering can be done in three ways, based on a user's context:

- within the retrieved set of documents, reordering those that have been given the same score
- within the retrieved set of documents, reordering all of them
- retrieving a completely new set of relevant documents

The first method of reordering was used in this work. Future work will test the effect of the others too. However, initially a minimal interventionist approach was preferred. Documents having the same score are said to belong to the same Score-Block (or Weight-Block).

After each query with positive relevance feedback, the context learner updates the user's context. As part of module D, the terms in the context are either put on hold (H) for possible use in subsequent iterations or they are put in use (U). The U terms affect the document ordering. They are used to break-up the score-blocks, i.e. reorder the documents within score-blocks. A query comprising all use (U) terms in the context is submitted to the probabilistic IRS, and the ordering obtained over all documents is applied to the documents within a score-block.

The incremental context learner was incorporated into the standard probabilistic IRS (Okapi) as follows:

```
FOR each user query DO {
    Parse query.
    If not first query on system
        (i.e. Context description exists) then
        use Context.
    EndIf
    Show ranked document list,
        with brief document descriptions.
    If user chooses to see more detailed
    information about document) then
        prompt user for a relevance judgement, before
        moving on to anything else.
    EndIf
    If there are any positive relevance judgements then
        update Context.
    EndIf
}
```

5 Evaluation of the Learner

Experiments were necessary not only to test the effectiveness of the learning algorithms but also to clarify aspects relating to the notion of context and its role in document ranking. Two experiments were performed for this.

Their purpose was to explore certain aspects such as: Is a context C_n derived after n queries useful for the subsequent query Q_{n+1}? Are these two (C_n and Q_{n+1}) related, strongly or weakly? Is context something that is constant or is it changing? What is the role and effect of minimum coverage? What is the role of term frequency (in relevant documents) in identifying context? Does document ordering, particularly within the same score-block, improve with the use of context?

The first experiment was performed to decide amongst different versions of the CL incorporated into the standard IRS. The various learning algorithms in the CLs represented theoretically differing viewpoints regarding the notion of context. As a result of Experiment 1, candidate CLs were identified for the next experiment.

The second experiment was carried out in order to compare how the candidate CL versions performed in relation to the standard version of the Okapi probabilistic IRS. The purpose of this experiment was to establish whether incorporating any of the candidate CLs, as determined from the previous experiment, improved document ordering.

The first of the hypotheses described in Section 3 is effectively tested in both experiments. If context is stable, then one would expect the CLs employing algorithms based on a use of R_{total} (module A) to perform consistently better than those with R_{latest}. In this case, the greater the number of context learning iterations, the more accurate the approximation to context should be.

The second hypothesis is tested by the second experiment where the candidate CL versions are compared against the performance of the standard system (the IRS without learning). If any one of the CLs perform better than the standard IRS, then the use of context improves document ordering. How much it improves document ordering has to be analysed by comparing the performance of each CL with the standard IRS.

Experimental Setup

Both experiments used data from different frequent users of the IRS. Those using the system in the last 3 months at least 5 times were identified as being frequent users. There were 108 queries over 63 sessions for 11 frequent users used for the first and 102 queries over 57 session for 9 different users for the second experiment.

The relevance assessments in both experiments were gathered for users' last queries on the system and they were approached to make these judgements within the week of performing it. Prior to this, they were not aware of these particular experiments and the participation required from them specifically.

The following sections describe a method for choosing amongst the CL variations and then comparing those chosen against the standard version of the system.

5.1 Measuring Effectiveness of a Learner

The users' relevance judgements, obtained for evaluation, consisted of three categories: Relevant (R), Partially relevant (P) and Non-relevant (N). They were asked to make these judgements based on the title, author, abstract, subject headings, source, and publication fields for a document record. They were instructed to make the judgements irrespective of whether they had actually seen the document before.

The results for both experiments consist of precision values (the proportion retrieved documents relevant) generated from the Relevant, Partially-relevant, Non-relevant figures in the users' relevance judgements. Precision is one of the standard measurements used in information retrieval. Another standard measurement is recall (the proportion of relevant documents retrieved). However, calculating recall is not possible as (unlike in test collections) there is no figure for the total number of documents relevant in the database for a particular query.

Various *cut-off points* (5, 10, 15, 20) were used for analysing the precision at certain rank positions in the document list i.e. how many of the first 5 documents are relevant and so on. In the absence of recall values, this approach helps identify some trends in relevance e.g. how many items are relevant near the top of the list compared to those further down the list.

5.2 Choosing amongst Competing Algorithms

This experiment was essentially a preliminary stage in order to perform Experiment 2. There were 14 CL versions and an evaluation based on users' assessments for all CLs is not feasible from the users' point of view. The reason being that each user would be expected to assess at least 140 documents - assuming they are given at least 10 for each CL. Therefore, users were asked to assess the results of the standard system. They were asked relevance judgements on the top 20 documents, for the last query, presented by the standard version of the IRS. Although the document lists produced by the standard system were used as a basis, the aim was not to compare with Okapi, for that Experiment 2 was devised. At this stage, CL versions were compared amongst each other.

The precision values for each learner (over all users) at cut-off points 5, 10, 15 and 20 were obtained. Two precision values are observed: P_R for relevant (R) documents; P_R for relevant together with partially relevant (R+P) documents. This is to cover for a spectrum of relevance judgements.

Thus, each learner has two precision values at each cut-off point. For this purpose, we can refer to a 'slot' as all precision values of the same type (either P_R or P_{R+P}) for a particular cut-off point. For each slot, the highest precision value was identified and those values within the top 5% (and including the highest value) were marked. For example if the highest precision value amongst the context learners, calculated by P_{R+P}, for rank position 10 is 88% then all those between (and including) 83-88% are considered for that slot. The process is applied independently for each slot. Table 1 shows the slots and the respective range of percentage values considered.

In Table 2, the precision values obtained for a context learner (CL3) are shown. In this case, only two slots had a precision value within the top 5%. At cut-off 15, precision P_R of 79% falls within the range of values considered. Likewise, at cut-off 20, the value for precision P_R is within the range of values considered for that slot.

Table 1: The range of percentage values considered for each rank position (the cut-off point) for the two types of precision values, for Experiment 1.

Rank Position	P_R (%)	P_{R+P} (%)
5	54 - 57	85 - 90
10	47 - 52	83 - 88
15	45 - 50	77 - 82
20	42 - 48	79 - 84

Table 2: Precision values at various cut-off points for a Context Learner CL3 for Experiment 1.

Rank Position	P_R(%)	P_{R+P}(%)
5	44	77
10	38	81
15	42	**79**
20	**45**	72

The algorithms which scored within the top 5% for at least half the slots were identified as candidates for the second experiment, described in the next section.

5.3 Comparing with the Standard System

Having performed Experiment 1, the next stage was to establish whether any of the four candidate CLs determined by the previous experiment improved (preferably significantly) the standard version of the system originally installed. In otherwords, if users judged a higher proportion of documents to be relevant (for the last query) in comparison with the standard IRS, then it constitutes an improvement.

In this experiment, users evaluated the (top 10) documents produced by each remaining CL and the standard IRS - a maximum of 50 documents. In practice, because some documents were retrieved by more than one CL version, there were overlaps which resulted in 30-40 documents per user. The precisions for these were calculated at various cut-off points and the highest ones identified.

Table (3) below shows precision values for the best and worst performing CL versions and the standard IRS. The worst CL does uniformly worse than the standard IRS, and the best CL does uniformly better than the standard IRS. In addition, the improvement is greatest when the precision values relating to partially-relevant documents are taken into consideration.

Using R_{total} (in Module A) with no overall thresholds when merging with the old context (Module C) produces worse results than R_{latest}. This would suggest that context does change over time. Using R_{total} with too high a threshold produces similarly poor results because the approximation adapts too slowly over time.

Table 3: The precision values at cut-off points for the standard IRS, best and worst performing CL

		System					
		Standard IRS		Worst CL		Best CL	
	Rank Pos.	5	10	5	10	5	10
Prec. (%)	P_R	42	32	38	28	47	44
	P_{R+P}	71	68	60	58	80	77

5.4 Discussion of Results

In summary, findings presented above indicate that there does appear to be a context for users' queries. Although it is not constant, it does not appear to change too quickly either (Section 5.3). This indicates it is worth pursuing context learning on a batch of consecutive queries for a user.

In the first experiment, each CL version using minimal coverage (described in Section 4.2) performed well in only one slot. Thus, they did not get tested in the second experiment. The approach of minimally covering the set of relevant documents to approximate context does not provide sufficiently acceptable results. It appears that too much information is lost when aiming for this type of coverage.

Ideally all CLs would be evaluated using the method in the second experiment, described in Sections 5 and 5.3. However due to the reasons given in Section 5.2, a process of elimination was applied to the CL versions.

Regarding the candidate CLs in the second experiment (Section 5.3), the results indicate that context is relatively stable. In order to reflect the continuity between the iterative contexts for a user, the proportion of terms in the context that changeover is very important.

6 Future Work

A context learning approach and its associated user-based evaluation framework have been presented. Further experiments involving larger number of users are desirable. However, generally it is not usually possible to do very large-scale in-depth studies of individual users and a balance has to be made between the level of in-depth studies of users and the number of users. It would also be appropriate to test the CL within the Web environment.

Finally, provided that there is an overlap between individual users' contexts, this approach could be extended to perform context learning on multiple users. Thus, the extent to which a CL can be used for collaborative learning or filtering over multiple users' contexts should also be investigated.

7 Acknowledgements

The author is grateful to David Harper and Robin Boswell for their helpful comments and suggestions on this paper.

References

[Armstrong *et al.*, 1997] R. Armstrong, D. Freitag, T. Joachims and T. Mitchell. WebWatcher: A learning apprentice for the World Wide Web. *Machine Learning and Data Mining: Methods and Applications* Ed. by R.S. Michalski, I. Bratko and M. Kubat, John Wiley & Sons Ltd., 1997.

[Belkin *et al.*, 1982] N. Belkin, R. Oddy, and H. Brooks. ASK for information retrieval. *Journal of Documentation*, 1982, 38, pp.61-71 (Part I); pp. 145-164 (Part 2).

[Bloedorn *et al.*, 1996] E. Bloedorn, I. Mani, and T.R. MacMillan. Representational issues in machine learning of user profiles. *AAAI Spring Symposium on Machine Learning in Information Access*, 25-27 March, 1996, Stanford University, CA.

[Chalmers *et al.*, 1998] M. Chalmers, K. Rodden, and D. Brodbeck. The order of things: Activity-centered information access. *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, Brisbane, 1998, pp. 359-367.

[Goker and McCluskey, 1991] A. Goker and T.L. McCluskey, Towards an adaptive information retrieval system. In: *Methodologies for Intelligent Systems. 6th International Symposium, ISMIS '91, Proceedings*, Springer-Verlag, 1991, ed by Ras, Z W; Zemankova, M, pp. 348-57.

[Jaczynski and Trousse, 1998] M. Jaczynski, and B. Trousse. WWW assisted browsing by reusing past navigations of a group of users. *European Workshop on Case-based Reasoning*, 1998. pp. 160-171.

[Jansen *et al.*, 1998] B.J. Jansen, A. Spink, J. Bateman, T. Saracevic. Real Life Information Retrieval: A study of user queries on the Web. *SIGIR-Forum*, Spring 1998. vol32 No 1. pp. 5-17.

[Robertson, 1997] S. E. Robertson. Overview of the Okapi projects. *Journal of Documentation* vol. 53, no. 1, 1997, pp. 3-7.

[Walker and Hancock-Beaulieu] S. Walker, and M. Hancock-Beaulieu. Okapi at City: An evaluation facility for interactive IR. *British Library R & D Report* No. 6056, 1991.