Incremental Methods for Context-Based Web Retrieval

Carlos M. Lorenzetti[§] **Fernando M. Sagui**[§] **Ana G. Maguitman**[§]
Carlos I. Chesñevar[§‡] **Guillermo R. Simari**[§]
e-mail: {cml,fms,agm,cic,grs}@cs.uns.edu.ar
[§]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, (8000) Bahía Blanca, Argentina[*]

[‡] Department of Computer Science
University of Lleida
C/Jaume II, 69 – 25001 Lleida, Spain[†]

Abstract

Intelligent search depends on effective methods for identifying the information needs of a user and making relevant information resources available when needed. Reflecting user context has long been recognized as a key aspect to realizing the potential of intelligent Web search. This paper proposes a theoretical basis for better understanding the role of context in Web retrieval. It addresses the problem of identifying context-specific terms, finding relevant information sources, and automatically formulating and refining queries. We describe ongoing research on the use of *incremental methods* to retrieve relevant content through two main approaches. The first, *feed-based*, periodically checks for new relevant items in specific websites by accessing RSS feeds. The second, *query-based*, incrementally formulates queries, which are submitted to search interfaces (e.g., major search engines or individual search forms). We discuss the technical challenges imposed by these approaches, outline our system architecture, and present preliminary evaluations of the proposed techniques.

Keywords: web search, context, RSS feeds, query formulation, incremental search

1 INTRODUCTION

The effectiveness of retrieval tools depends on their ability to anticipate the information needs of a user and automatically present useful resources to the user at the right time. The World Wide Web is an ever-expanding source of information about a huge diversity of topics. As a consequence, it is becoming increasingly important to know how to find out about a topic of special interest, focusing the search on material that is relevant to the current task. This search activity could be done more effectively if “intelligent mechanisms” for information access and delivery were included as part of the system search tools. In order to reduce the user cognitive overload, task-specific Web search tools need to be adapted to deliver few but highly relevant resources.

An important requirement for these tools is to provide relevant material, doing it at the right time, and without causing undue or excessive distraction. Two elements that can be exploited to enhance

[*] Partially supported by CONICET (PIP 5050), Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002, Nro. 13096) and Universidad Nacional del Sur (24/N016).

[†] Partially supported by TIN2006-15662-C02-02, Ministerio de Educación y Ciencia, Spain.

Web search are *user context* and *user preferences*. User context reflects the task in which the user is immersed (e.g., [4, 9]). The context may consist of an electronic document the user is editing, Web pages the user has recently visited, etc. User preferences reflect the way in which a user would prioritize search results. User preferences could be entered explicitly by the user or could be inferred by the system (e.g., by monitoring the user's behavior).

As part of our research work we are studying how to build intelligent aides that can provide context-based and preference-based support by retrieving useful information from the Web. These aides monitor the user and search the Web for material related to the user current task and preferences. A general discussion of the proposed architecture and project goals can be found in [5].

This paper reports advances in addressing the problem of context-based Web retrieval. In order to deal with this problem, we distinguish two general mechanisms of content retrieval from the Web:

- **Feed-based**: the system periodically checks for new relevant items by accessing RSS feeds in specific websites. A set of sites publishing feeds are specified in advance by the user, while the system is in charge of identifying and delivering potentially relevant material available through such feeds.

- **Query-based**: the system automatically formulates search queries, which are submitted to major search engines (e.g., Google) or entered into individual search forms (e.g., PubMed or Amazon). In both cases the queries can be incrementally refined based on initial results and their similarity to the user context.

Although the above approaches differ in various aspects and each has to face unique technical challenges, both need to address a major common issue, namely the identification of context-specific terms to guide the search. This paper presents general techniques for the incremental identification of important terms in the context of a user task. Specifically, we are studying three questions: (1) can the user context be usefully exploited to access relevant material?, (2) are those terms that occur more frequently in the user context sufficient to retrieve useful information?, and (3) can the set of context-specific terms be incrementally refined, based on the analysis of search results? To address these questions section 2 presents a theoretical framework for the study of contextual search on the Web. Section 3 proposes incremental methods for Web retrieval and discusses the technical challenges faced by the feed-based and query-based retrieval approaches. Section 4 shows the results of our initial evaluations, and finally, section 5 presents our conclusions.

2 A GENERAL FRAMEWORK FOR CONTEXTUAL SEARCH

Search interfaces provide access to a vast repository of information on the World Wide Web. However, finding relevant information remains challenging, because of the need to select useful resources from an enormous range of possibilities. For many computer-mediated tasks, the user context provides a rich set of terms that can be exploited to enhance Web search. A context-based search tool can be embedded in different kinds of computer utilities, such as email systems, browsers and text editors. In order to characterize the user task it is necessary to bring into play a set of context-specific terms. This requires a framework for weighting terms based on context.

2.1 Term Importance in Classical IR

Substantial experimental evidence supports the effectiveness of using weights to reflect relative term importance for traditional information retrieval (IR) [14]. The main purpose of a term weighting system is the enhancement of retrieval effectiveness.

Effective retrieval depends on retrieving those items that are likely to be relevant to the user's needs, but also on filtering irrelevant material. In order to assess the ability of a system to retrieve relevant items and reject the irrelevant ones, the IR community normally uses two measures, known as *recall* and *precision*.

Given an information request and its set of relevant documents R, assume that a given retrieval strategy generates a document answer set A. The recall and precision measures are defined as follows [2]:

- **Recall** is the fraction of relevant documents (the set R) which has been retrieved, i.e.,

$$\textbf{Recall} = \frac{|R \cap A|}{|R|}$$

- **Precision** is the fraction of retrieved documents (the set A) which is relevant, i.e.

$$\textbf{Precision} = \frac{|R \cap A|}{|A|}$$

Note that the recall measure, as defined above, assumes that we have access to $|R|$, the number of relevant documents. For a large and dynamic corpus, such as the Web, it is impossible to determine this number. However, approximations for the recall and precision measures for the Web domain have been proposed in a number of studies (e.g, [15, 7, 17]).

In principle, a system is preferred if it produces both high recall and high precision. To serve recall and precision, conventional IR schemes use composite term weighting factors that contain both recall- and precision-enhancing components. For several decades the IR community has investigated the role of terms as descriptors and discriminators. The combination of descriptors and discriminators gives rise to schemes for measuring term importance such as the familiar *term frequency inverse document frequency* (TF-IDF) weighting model [14]. TF-IDF is a simple way to measure the relevance of a term for a document relative to a collection. According to the TF-IDF scheme, term relevance is determined by two quantities:

- **Term Frequency**. Given a document d and a term t, the *term frequency* is simply measured as the number of times term t occurs in document d:

$$\textit{TF(d,t)} = n(d,t)$$

- **Inverse Document Frequency**. Given a term t and a collection D of documents, the *inverse document frequency* measure varies inversely with the number of documents to which a term is assigned. In its common form, *inverse document frequency* is defined as follows [14]:

$$\textit{IDF(t)} = \log \frac{1 + |D|}{|D_t|}$$

  where $|D_t|$ represents the number of documents in D containing term t.

Term frequency factors help to achieve high recall. However, term frequency alone cannot insure acceptable precision because high frequency terms may also occur in irrelevant documents. Hence inverse document frequency performs the function of penalizing those terms that lack discriminating power. TF and IDF are combined to form the TF-IDF measure as follows:

$$\textit{TF-IDF(d,t)} = \textit{TF(d,t)} \times \textit{IDF(t)}$$

The TF-IDF scheme is a reasonable measure of term importance but is insufficient for the task domain for our research. As has been discussed by a number of sources, issues arise when attempting to apply conventional IR schemes for measuring term importance to systems for searching Web data [8, 3]. One difficulty is that methods for Web search do not have access to a fully predefined collection of documents, raising questions about the suitability of classical IR schemes for measuring term importance when searching the Web. A central question addressed in our work is how to formulate topic descriptors and discriminators to guide context-based topic search on the Web.

2.2 Incremental Methods

Searching the Web presents a new challenge for the formulation of topic descriptors and discriminators. Assume a topic is represented as a set of cohesive terms summarizing the topic content. In previous work [12] we have tested the following two hypotheses:

- Good topic descriptors can be found by looking for terms that occur often in documents similar to the given topic.
- Good topic discriminators can be found by looking for terms that occur only in documents similar to the given topic.

To compute descriptive and discriminating power we begin with a collection of m documents and n terms. Each of these documents could be represented by an event in an RSS feed (as will be described later in section 3.1), or a snippet in a list of results returned by a search engine (as described in section 3.2), or could be a full text document, depending on the task at hand. As a starting point we build an $m \times n$ matrix $\mathbf{H}$, such that $\mathbf{H}[i, j] = k$, where k is the number of occurrences of term t_j in document d_i.

If we adopt $\mathrm{s}(k) = 1$ whenever $k > 0$ and $\mathrm{s}(k) = 0$ otherwise, we can define the *discriminating power of a term in a document* as a function $\delta : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \to [0, 1]$:

$$\delta(t_i, d_j) = \frac{\mathrm{s}(\mathbf{H}[j, i])}{\sqrt{\sum_{k=0}^{m-1} \mathrm{s}(\mathbf{H}[k, i])}}.$$

Analogously, we define *descriptive power of a term in a document* as a function $\lambda : \{d_0, \ldots, d_{m-1}\} \times \{t_0, \ldots, t_{n-1}\} \to [0, 1]$:

$$\lambda(d_i, t_j) = \frac{\mathbf{H}[i, j]}{\sqrt{\sum_{k=0}^{n-1} (\mathbf{H}[i, k])^2}}.$$

Note that δ and λ satisfy the conditions

$$\sum_j (\delta(t_i, d_j))^2 = 1 \quad \text{and} \quad \sum_j (\lambda(d_i, t_j))^2 = 1.$$

Given a term t_i in a document d_j, the term t_i will have a high discriminating power if it tends to occur only in d_j (i.e., it seldom occurs in other documents), while it will have a high descriptive power in d_j if it occurs often in d_j. These simple notions of document descriptors and discriminators share the same basic objective with the classical weighting schemes discussed in section 2.1. However, in the same way as TF and IDF, the functions λ and δ allow to discover terms that are good descriptors and discriminators of a document, as opposed to good descriptors and discriminators of the *topic* of a document.

Our current goal is to formulate notions of topic descriptors and discriminators suitable for the Web scenario. Rather than extracting descriptors and discriminators directly from the user context, we want to extract them from the topic of the user context. This requires an incremental method to characterize the topic of the user context, which is done by identifying documents that are similar to the user current context. Assume the user context and the retrieved content are represented as document vectors in term space. To determine how similar two documents d_i and d_j are we adopt the IR cosine similarity [2]. This measure can be computed as follows:

$$\sigma(d_i, d_j) = \sum_{k=0}^{n-1} [\lambda(d_i, t_k) \cdot \lambda(d_j, t_k)].$$

As we informally formulated earlier, a term is a good discriminator of a topic if it tends to occur *only* in documents associated with that topic. We define the *discriminating power of a term in the topic of a document* as a function $\Delta : \{t_0, \ldots, t_{n-1}\} \times \{d_0, \ldots, d_{m-1}\} \to [0, 1]$ calculated as follows:

$$\Delta(t_i, d_j) = \sum_{\substack{k=0 \\ k \neq j}}^{m-1} [[\delta(t_i, d_k)]^2 \cdot \sigma(d_k, d_j)].$$

Thus the discriminating power of term t_i in the topic of document d_j is an average of the similarity of d_j to other documents discriminated by t_i.

The notion of topic descriptors was informally defined earlier as terms that occur *often* in the context of a topic. We measure *term descriptive power in the topic of a document* as a function $\Lambda : \{d_0, \ldots, d_{m-1}\} \times \{t_0, \ldots, t_{n-1}\} \to [0, 1]$. If $\sum_{\substack{k=0 \\ k \neq i}}^{m-1} \sigma(d_i, d_k) = 0$ then we set $\Lambda(d_i, t_j) = 0$. Otherwise we compute $\Lambda(d_i, t_j)$ as follows:

$$\Lambda(d_i, t_j) = \frac{\sum_{\substack{k=0 \\ k \neq i}}^{m-1} [\sigma(d_i, d_k) \cdot [\lambda(d_k, t_j)]^2]}{\sum_{\substack{k=0 \\ k \neq i}}^{m-1} \sigma(d_i, d_k)}$$

The descriptive power of a term t_j in the topic of a document d_i is a measure of the quality of t_j as a descriptor of documents similar to d_i.

Guided by the notions of topic descriptors and discriminators, it is possible to reinforce the weights of existing and novel context-specific terms. This results in a better representation of the user search context, facilitating query refinement and context-based filtering.

3 APPLYING THE FRAMEWORK IN THE IMPLEMENTATION OF CONTEXT-BASED RETRIEVAL SYSTEMS

Incremental search methods are useful for collecting information from diverse information sources on the Web. The incremental identification of context-specific terms can guide the search process through huge repositories of potentially useful material, helping to filter irrelevant content.

The implementation of an incremental context-based Web retrieval system requires several specialized components. Figure 1 presents a general architecture for an incremental context-based search system. This architecture includes the following modules:

- **Web Interaction Module**. This component handles the communication with the World Wide Web. It is in charge of contacting remote information sources, retrieving content from these sources, and pre-processing the retrieved content.

- **Context-Based Filtering Module**. This module is in charge of estimating the relevancy of the content collected by the Web interaction module. This is done by computing the similarity between the collected material and the user current context. Both relevant and irrelevant material will be kept for use by the incremental context-refinement module (described below). However, only the material that is relevant to the current context will be presented to the user.
- **Incremental Context-Refinement Module**. This module uses the content returned by the Web interaction module in combination with the relevancy information provided by the context-based filtering module to incrementally refine the context representation. It does so by adjusting the weights of the context-specific terms according to their descriptive and discriminating power.

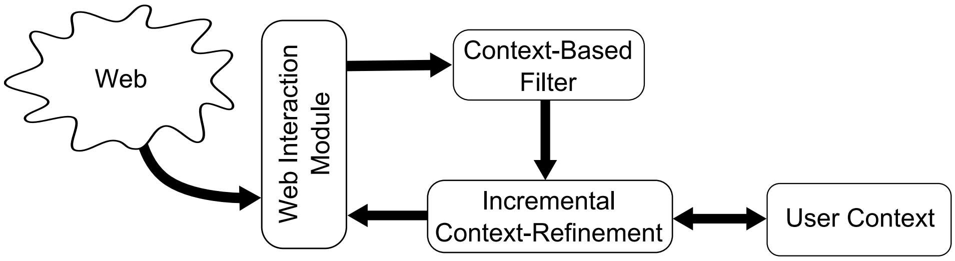


Figure 1: General architecture for an incremental context-based search system on the Web

The methods proposed in section 2.2 play an important role in the design of the incremental context-refinement module. An incremental approach to identify context-specific terms allows to go beyond the known user desires, to automatically generate a richer context representation through the use of topic descriptors and discriminators, and find what might be useful for the user. This kind of incremental mechanism can reveal similarities that were not previously apparent and present a "big picture" that can give the user a broader understanding of the current task.

We are concerned with two particular instances of context-based Web search systems: feed-based and query-based. Retrieving context-relevant information from the Web through feeds or by querying search interfaces poses specific technical challenges. In the following two sections we briefly review RSS-feed and search-interface technologies and discuss a set of methods used to address the challenges brought by each of them.

3.1 Retrieving Information Through RSS Feeds

RSS is an abbreviation for "Really Simple Syndication" and is concerned with syndicating ephemeral content such as news headlines and users' blog entries. RSS feeds are dynamically generated summaries (in XML format) of information published on websites. This technology helps automate the process of getting a quick update of the current state of a website with simple titles and links. Because RSS feeds are rich in metadata annotations they can be easily processed by systems.

RSS readers periodically look at the information available through these feeds to signal changes and trigger new actions. These mechanisms allow diverse applications to interact, resulting in a powerful networked infrastructure for the efficient organization and distribution of content. Because of these characteristics, RSS technology is particularly suitable to deal with frequently changing material and has been successfully applied to distribute information in social environments such as blogs and wikis, support annotation tools (e.g., Flickr), empower hybrid utilities (e.g., JobSpot), publish top news from different sources (NYTimes, Clarin.com), and announce latest software releases.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<rss version="2.0"> - <channel>
  <title>NYT > Middle East</title>
  <link>http://www.nytimes.com/pages/world/middleeast/index.html?partner=rssnyt</link>
  <description />
  <copyright>Copyright 2006 The New York Times Company</copyright>
  <language>en-us</language>
  <lastBuildDate>Thu, 20 Jul 2006 20:05:00 EDT</lastBuildDate>
  <image>
  <url>http://graphics.nytimes.com/images/section/NytSectionHeader.gif</url>
  <title>NYT > Middle East</title>
  <link>http://www.nytimes.com/pages/world/middleeast/index.html</link>
  </image>
  <item>
   <title>Marines Aiding Evacuation in Beirut; New Clash in South</title>
   <link>http://www.nytimes.com/2006/07/20/world/middleeast/20cnd-mideast.html?
   ex=1311048000&en=01e2fa49353969e5&ei=5088&partner=rssnyt&emc=rss</link>
   <description>As fighting continued, Israeli officials suggested that ground
   troops may take a more active role. Stronger condemnations today were heard
   of Israel’s massive use of force in Lebanon.</description>
   <author>JAD MOUAWAD and STEVEN ERLANGER</author>
   <pubDate>Thu, 20 Jul 2006 00:00:00 EDT</pubDate>
   <guid isPermaLink="false">http://www.nytimes.com/2006/07/20/world/middleeast/
   20cnd-mideast.html</guid>
  </item>
  <item>
    ...
  </item>
  <item>
    ...
  </item>
</channel>
</rss>
```

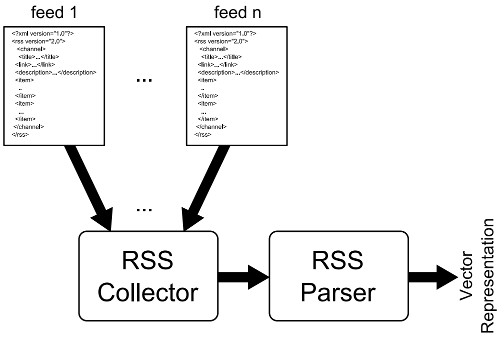
Figure 2: An RSS feed from the NYT newspaper

In figure 2 we present an example of an RSS feed found at the website of the *New York Times* newspaper.[1] Every feed has at least one *channel* element, containing metadata associated with the RSS resource, such as <*title*>, <*link*>, <*description*>, <*language*>, etc. In addition, it contains a list of items or events. Each of these events is typically characterized by a *title*, a *url* (containing a link to a full article), a *description* (containing a short summary of the event), an *author*, and a *publication date*.

The architecture outlined in figure 1 can be naturally adapted to deal with RSS feeds. Collecting material from feeds requires the implementation of a specialized Web interaction module, which we outline in figure 3. The Web interaction, in this case, is done through the following components:

- **RSS Collector**. This component is in charge of periodically visiting and retrieving RSS feeds from different locations. It handles all communication problems such as timeouts, availability and connectivity.
- **RSS Parser**. This module parses the XML file associated with a particular RSS channel. Guided by the metadata annotations within a feed, it extracts the key elements and builds a local representation of each event found in the feed. Each event is represented as a document vector in term space.

[1] http://www.nytimes.com/



```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
   <title>...</title>
 <link>...</link>
 <description>...</description>
 <item>
   ..
 </item>
 <item>
   ...
 </item>
 </channel>
</rss>
```

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
   <title>...</title>
 <link>...</link>
 <description>...</description>
 <item>
   ..
 </item>
 <item>
   ...
 </item>
 </channel>
</rss>
```

Figure 3: Web interaction module for a feed-based search system

```
Israel Invasion of Lebanon in 1978
Palestine Facts is a review of the historical, political and military facts
behind the State of Israel and the Israeli-Arab Palestinian conflict.
www.palestinefacts.org/pf_1967to1991_lebanon_1978.php - 20k - Cached - Similar pages

Israel’s Lebanon Policy
Although the conflict with Hizballah now dominates Israel’s Lebanon
problem, ... One comprehensive overview of Israel’s woes in Lebanon
asserts that the ...
meria.biu.ac.il/journal/1997/issue3/jv1n3a3.html - 42k - Cached - Similar pages

...

Israel, Lebanon, and the "Peace Process", by Noam Chomsky
Lebanon has been a victim of the Arab-Israel conflict for half a century. In
1948, and again in 1967, it was a dumping ground for Palestinians
who fled or ...
www.chomsky.info/articles/19960423.htm - 12k - Cached - Similar pages
```

Figure 4: Search results returned by Google for the query *israel lebanon*.

3.2 Retrieving Information Through Search Interfaces

Search interfaces provide fast access to information available on the Web. Differently from accessing information through RSS feeds, which require sequential access to content, the query-based approach can take advantage of information indexed by major search engines (e.g., Google) or other searchable databases (PubMed, Amazon, etc.). The major challenge that the query-based approach needs to address is the generation of suitable queries.

Users often generate very short queries (a study of over a million queries to the Excite search engine showed that 60% of queries were one or two terms long [16]), making query ambiguity a serious problem. Users may be inexperienced in selecting suitable keywords, may not know enough about the domain to select good query terms, or may simply overlook useful keywords. Some research addresses these problems by automatically augmenting user queries based on the task context, e.g., to make related suggestions as users read or write a document (e.g. [4]).

Our methods focus on how to incrementally generate queries or augment user queries based on context. Because search engines restrict queries to a small number of terms (e.g., the 32-term limit for Google) a single query cannot reflect extensive contextual information. In an incremental method, the first query terms generated for a Web search may not provide the definitive results. However, comparing the set of search results to the user context can help to automatically refine subsequent queries.

Figure 4 shows a set of results returned by Google for the query *israel lebanon*. We use Google Web API to collect search results and only the “snippets” returned by Google are used by our methods. The snippet is a text excerpt from the page summarizing the context in which the search terms occur.

Figure 5 depicts the necessary components for the implementation of a Web interaction module through a query-based search interface:

- **Query Generator**. This component selects terms from the user context and forms suitable queries, which are submitted to a standard search engine (e.g, Google) or entered into individual search forms (e.g., Amazon or PubMed). Initial queries will be entered by the user, or automatically formed with terms that occur frequently in the user context. Subsequent queries are refined as topic descriptors and discriminators are identified by the incremental search method.

- **Results Retrieval Module**. This component is in charge of retrieving the search results generated by the search interface, so that they can be locally analyzed and transformed to vector representations in term space.

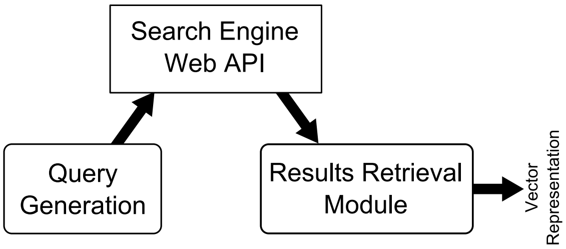


Figure 5: Web interaction module for a query-based search system

4 PRELIMINARY EVALUATION

Because the goal of the retrieval process is to present useful suggestions, the ideal method for evaluating result quality would be an end-to-end user study, in which subjects directly assess the usefulness of system suggestions. However, to guide the bottom-up development of the methods, it is crucial to be able to assess incremental steps for which human-subjects evaluations would be impractical.

In previous work we showed that topic descriptors can help achieve good recall, while topic discriminators improve precision [12, 11]. Here, we present new empirical evidence that points out to the usefulness of combining topic descriptors and discriminators to maximize the contribution of both to form suitable queries. An evaluation of the feed-based approach will be addressed in future work.

In our evaluations we start from a TF vector representation C of a base text (representing the user context), and parameters l, m and n. The incremental method generates queries as follows:

1. Generate the initial Query(0) using the l terms in C with highest TF;
2. i=0;
3. Send Query(i) to the search engine;
4. Obtain the answers set and convert the results to a vector representation;
5. Generate a sorted list L_Λ of topic descriptors;
6. Generate a sorted list L_Δ of topic discriminators;
7. i= i +1;
8. Query(i) $\leftarrow$ some combination of m terms from L_Λ and n terms from L_Δ;
9. go to3;

The input parameter l in the above algorithm determines the initial query size. The parameters m and n specify the number of descriptors and discriminators, respectively, used to form each of the subsequent queries. Note that this method generalizes those in which only descriptors or only discriminators are used (since either m or n could be set to zero).

We illustrate the effectiveness of the proposed method with an example that starts from a base text extracted from a page on the topic of "the problem of the petroleum and their imminent dearth". This document was obtained from Google by searching for the keyword "oil" and then advancing several pages forward until those with low pageranks were reached. Table 1 shows the list of most frequent terms in this document.

order	**term**	**TF (normalized)**
1	campaign	0.3834
2	oil	0.3067
3	public	0.3067
4	chevron	0.2300
5	truth	0.2300
6	propaganda	0.1533
7	peak	0.1533
8	confidence	0.1533
9	relations	0.1533
10	firm	0.1533

Table 1: Terms extracted directly from the initial context sorted by TF.

As a baseline for comparison we used a simple search mechanism that attempts to mimic the way in which human beings perform searches when looking for information on a particular topic. This baseline method generates queries by selecting the terms that occur with most frequency in the search context.

Tables 2 and 3 show the sequences of queries submitted by the baseline method and the incremental method, respectively. The initial query for the tested methods was based on TF, and therefore was identical in both cases, resulting in approximately 178000 hits. For subsequent queries, the terms in boldface are those that do not exist in the original text. It is also interesting to note that some of these new terms appear in preponderant positions, which can have an important impact on the set of results returned by search engines (e.g., Google takes into consideration the position of keywords in queries).

campaign oil public chevron truth
oil chevron campaign **click dependence**
oil campaign chevron **dependence reduce**

Table 2: Sequence of queries submitted by the baseline method.

campaign oil public chevron truth
oil campaign confidence **crash** future goal **harsh life look** maintain
oil **harsh leg2capital** chevron **look simply** truth **crash** campaign maintain

Table 3: Sequence of queries submitted by the incremental method.

Figure 6 shows the minimum, average and maximum similarities between the search results (snippets) and the initial context (base document), for the three iterations in the above example. Preliminary tests, as illustrated by this example, indicate that the incremental method outperforms the baseline. However, rigorous evaluations are still underway. In particular we are interested in performing additional tests to understand the effect of query size on retrieval performances as well as the use of special query syntaxes to combine descriptors and discriminators.

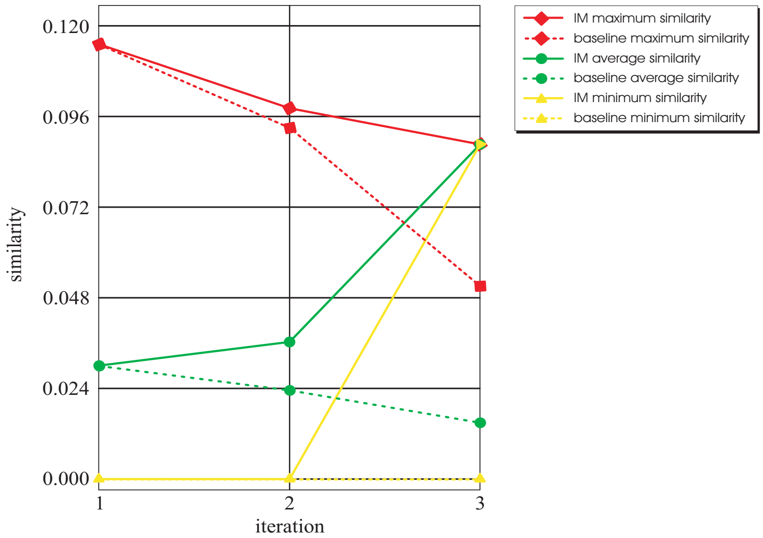


Figure 6: A comparison between the incremental method (IM) and the baseline method

5 CONCLUSIONS

The use of context to select and filter information plays a vital role in proactive retrieval systems. Such systems observe user interactions, infer user needs for additional information resources, and search for relevant documents on the Web or other online electronic libraries. For example, Watson [4] uses contextual information from documents that users are manipulating to automatically generate Web queries from the documents, using a variety of term-extraction and weighting techniques to select suitable query terms. Watson then filters the matching results, clusters similar HTML pages, and presents the pages to the user as suggestions. Another such system is the Remembrance Agent [13] which operates inside the Emacs text editor and continuously monitors the user's work to find relevant text documents, notes, and emails previously indexed. Other systems such as Letizia [10] and WebWatcher [1] use contextual information compiled from past browsing behavior—searches within the locus of a currently viewed Web page—to provide suggestions on related Web pages or links to explore next.

This paper has described ongoing research on taking advantage of the information available in the user context to perform incremental search on the Web. We have shown that the user context can be usefully exploited to access relevant material. However, those terms that occur more frequently in the user context are not necessarily the most useful ones. In light of this we proposed an incremental method for context refinement based on the analysis of search results.

Incremental methods are crucial for the effective retrieval of novel but useful material in frequently changing domains, such as news feeds. Preliminary evaluations also show the effectiveness of incremental methods for query generation and refinement. We are currently working on integrating the proposed method with qualitative approaches such as the ones discussed in [5, 6] for ranking results based on user preferences.

REFERENCES

[1] R. Armstrong,D. Freitag, T. Joachims, T. Mitchell. WebWatcher: A learning apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering* pages 6–12, AAAI Press, 1995.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[3] N. J. Belkin. Helping people find what they don't know. *Commun. ACM*, 43(8):58–61, 2000.

[4] J. Budzik, K. J. Hammond, and L. Birnbaum. Information access in context. *Knowledge based systems*, 14(1–2):37–53, 2001.

[5] C. Chesñevar, C. Lorenzetti, A. Maguitman, F. Sagui, and G. Simari. In *Proceedings of the Workshop de Investigadores en Ciencias de la Computación (WICC 2006)*, May 2006.

[6] C. Chesñevar and A. Maguitman. ArgueNet: An argument-based recommender system for solving Web search queries. In *Proceedings of the International IEEE Conference on Intelligent Systems (IS 2004)*, pages 282–287. IEEE, June 2004.

[7] H. Chu and M. Rosenthal. Search engines for the World Wide Web: A comparative study and evaluation methodology. In *Annual Conference Proceedings (ASIS'96)*, pages 127–135, October 1996.

[8] M. Kobayashi and K. Takeda. Information retrieval on the Web. *ACM Computing Surverys*, 32(2):144–173, 2000.

[9] D. B. Leake, T. Bauer, A. Maguitman, and D. C. Wilson. Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In *Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems*, pages 33–37. AAAI Press, 2000.

[10] H. Lieberman. Letizia: An agent that assists Web browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, San Mateo, Morgan Kaufmann, 1995.

[11] A. Maguitman, D. Leake, and T. Reichherzer. Suggesting novel but related topics: Towards context-based support for knowledge model extension. In *Proceedings of the IUI Conference*, pages 207–214, New York, January 2005. ACM Press.

[12] A. Maguitman, D. Leake, T. Reichherzer, and F. Menczer. Dynamic extraction of topic descriptors and discriminators: Towards automatic context-based topic search. In *Proceedings of the CIKM Conference*, pages 463–472, New York, November 2004. ACM Press.

[13] B. Rhodes and T. Starner. The remembrance agent: A continuously running automated information retrieval system. In *The Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology*, pages 87–495, London, UK, 1996.

[14] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372, 1973.

[15] T. Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of SIGIR*, pages 138–146, 1995.

[16] A. Spink, D. Wolfram, B. J. Jansen, and T. Saracevic. Searching the web: The public and their queries. *Journal of the American Society for Information Sciences and Technology*, 52(3):226–234, 2001.

[17] L. Wishard. Precision among Internet search engines: An earth sciences case study. Issues in Science and Technology Librarianship, Spring 1998.