Mining Topic-Specific Concepts and Definitions on the Web

Bing Liu
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan Street
Chicago, IL 60607-7053
liub@cs.uic.edu

Chee Wee Chin
Department of Computer Science
National University of Singapore
3 Science Drive 2
Singapore 117543
cheewee@alumni.nus.edu.sg

Hwee Tou Ng
Department of Computer Science
National University of Singapore
3 Science Drive 2
Singapore 117543
nght@comp.nus.edu.sg

ABSTRACT

Traditionally, when one wants to learn about a particular topic, one reads a book or a survey paper. With the rapid expansion of the Web, learning in-depth knowledge about a topic from the Web is becoming increasingly important and popular. This is also due to the Web's convenience and its richness of information. In many cases, learning from the Web may even be essential because in our fast changing world, emerging topics appear constantly and rapidly. There is often not enough time for someone to write a book on such topics. To learn such emerging topics, one can resort to research papers. However, research papers are often hard to understand by non-researchers, and few research papers cover every aspect of the topic. In contrast, many Web pages often contain intuitive descriptions of the topic. To find such Web pages, one typically uses a search engine. However, current search techniques are not designed for in-depth learning. Top ranking pages from a search engine may not contain any description of the topic. Even if they do, the description is usually incomplete since it is unlikely that the owner of the page has good knowledge of every aspect of the topic. In this paper, we attempt a novel and challenging task, *mining topic-specific knowledge on the Web*. Our goal is to help people learn in-depth knowledge of a topic systematically on the Web. The proposed techniques first identify those sub-topics or salient concepts of the topic, and then find and organize those informative pages, containing definitions and descriptions of the topic and sub-topics, just like those in a traditional book. Experimental results using 28 topics show that the proposed techniques are highly effective.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, query formulation*, *retrieval models*, *search process*, *selection process*.

General Terms

Algorithms, Experimentation

Keywords

Web content mining, domain concept mining, definition mining, knowledge compilation, information integration.

WWW 2003, May 20-24, 2003, Budapest, Hungary.
ACM 1-58113-680-3/03/0005.

1. INTRODUCTION

With the rapid expansion of the Web, the content of the Web is becoming richer and richer. People are increasingly using the Web to learn an unfamiliar topic because of the Web's convenience and its abundance of information and knowledge. It is even beginning to challenge the traditional method of learning. In traditional learning, if one is interested in learning a particular topic, one finds and reads a book or a survey paper on the topic. This classic method is inconvenient because buying (or borrowing) a book is time consuming, while information on the Web is only a click away. In many cases, this traditional method of learning may not even be applicable because in our fast changing world, many topics and technologies emerge constantly and rapidly. There is often not enough time for someone to compile all the existing knowledge and to write a book. Although reading research papers to learn such topics is possible, the intricacy of research papers is not always appropriate for non-researchers. Moreover, the focuses of research papers are usually narrow and they seldom discuss application issues. The Web, on the other hand, often contains intuitive descriptions and applications of the topics or technologies. Unlike a book, the Web also has great diversity. It can offer many different descriptions or discussions of the same topic, which is very helpful to a learner. Learning from the Web is thus natural and intuitive.

When one tries to learn about a new topic, one typically wants to know the following:

(1) Definitions and/or descriptions of the topic: For example, if one wants to learn about data mining, one first wants to know what the definition of data mining is and what it does.

(2) Sub-topics and/or salient concepts of the topic: One also wants to know what are the sub-topics and/or salient concepts of the topic. Following the data mining example above, one wants to know those important sub-topics, e.g., classification, association rule mining, clustering, sequential rule mining, etc. These sub-topics enable one to gain a more complete and in-depth knowledge of the domain. The sub-topics may lead us to (1) recursively.

In short, one would like to have the knowledge presented as that in a book, which has a table of contents of sub-topics, and each sub-topic points to its sub-subtopics and content pages. Thus, we also call our task, *compiling a book on the Web*. Note that in this paper, we use the terms *sub-topics* and *salient concepts* interchangeably as their classification can be subjective.

On the Web, the most commonly used tools for learning are the search engines (e.g., Google, Yahoo!, Altavista, and others). The user first submits a search query representing the topic to a search engine system, which finds and returns those related Web pages. He/she then browses through the returned results to find those suitable Web pages that contain knowledge of the topic. However, current search techniques are not designed for in-depth learning on the Web. Let us discuss why that is the case based on the two learning requirements above.

(1) Definitions and/or descriptions of the topic: Existing search engines rank Web pages mainly based on keyword matching and hyperlink structures (e.g., authorities and hubs) [7][20] [28]. Not enough attention has been paid to measure the *informative values* of Web pages (an *informative* page is a page that contains a definition and/or description of the topic). This results in many top-ranking pages containing no definition or description of the search topic. For example, we submit a search query "Web mining" to the Google search engine [7] hoping to learn about the topic. The first document returned by Google is http://www.kdnuggets.com/, which provides an excellent coverage of related resources. It is both a good authority and hub page. However, the page gives minimal explanation of what Web mining is. Thus, the page is more suitable for people who are already familiar with data and/or Web mining, and need additional resources. It is not a site designed for people who want to learn about Web mining.

(2) Sub-topics or salient concepts of the topic: Unlike a book or a good survey paper, a single Web page is unlikely to contain information about all the key concepts and/or sub-topics of the topic. This is due to the fact that the author of the page may not be an expert on every aspect of the topic and/or may not be interested in every aspect of the topic. Thus, sub-topics need to be discovered from multiple Web pages. Current search engine systems do not perform this task.

Due to these reasons, Web users often find it difficult to learn about an unfamiliar topic using search engines. Clearly, there is a need to develop novel techniques to overcome these impediments in order to help users learn on the Web easily. Note that such systems not only are able to help a learner who is unfamiliar with the topic, but are also able to help an expert who wants to compile the knowledge in the area for teaching purposes, and/or for writing a book or survey article on the topic.

By no means are we criticizing search engines. In fact, existing search engines are already extremely useful. However, the users of the search engines are not uniform. Presenting the same result to different types of users with diverse information needs (i.e., the one-size-fits-all approach) may not be appropriate. As in the above example, although the kdnuggets authority page is very useful, we should also provide the user with those informative pages and key concepts of the topic if he/she is interested in learning about the topic.

Despite the importance and usefulness of this task, limited work has been done in the past. Apart from search engines, work related to ours includes Web information extraction (e.g., wrappers [3][10][14][16], Web queries languages [8][26], user preferences [32], etc), definition finding [19] and question-answering [11][17][18][22] in information retrieval. In the next section, we will discuss these related works and show that they are not sufficient and/or appropriate for our task.

In this paper, we propose a set of effective techniques to perform the task of mining and organizing topic-specific knowledge on the Web. The process starts with a search query (given by the user) representing the topic. The system then collects the set of top ranking pages (top 100 pages in our experiments) returned from a search engine, and processes them further to discover those sub-topics or salient concepts of the search topic. Following that, it identifies those informative pages, which contain definitions or descriptions of the topic and sub-topics or concepts. This process can be performed recursively on the sub-topics, and so on. One of the difficult problems is the ambiguity of some concepts, which means that the concepts may have multiple meanings and/or may appear in different contexts. When such a concept is submitted to a search engine, the results returned are often irrelevant. We will propose an effective technique to deal with the problem.

Using the proposed technique, the user can quickly gain a comprehensive understanding of a topic without going through the ordeal of browsing through a large number of non-informative pages (which give little useful knowledge) returned by the search engine. Extensive experiments show that the proposed technique is able to perform the task very effectively.

This paper is organized as follows. In Section 2, we review the related work. Following that, in Section 3, the proposed technique is presented. Section 4 gives the architecture of our system. Section 5 evaluates the proposed technique and the system. Section 6 concludes the paper and discusses some future work.

2. RELATED WORK

Ever since the inception of the Web, searching and extracting useful information from it has been an active research area. So far, many information extraction techniques have been proposed and some of them are also widely used in practice. These techniques include keyword-based search, wrapper information extraction, Web queries, user preferences, and resource discovery. Keyword-based search using search engines (e.g., [7]) is clearly insufficient for our task as discussed in the Introduction section. Wrapper-based approaches (e.g., [3][10][14][16]) are not suitable either because Wrappers basically help the user extract specific pieces of information from targeted Web pages. Hence, they are not designed for finding salient concepts and definitions of user-specified topics, which can be of any type. Web query languages (e.g., [8][26]) allow the user to query the Web using extended database query languages. They are also not suitable for our problem. In the user preference approach (used commonly in push type of systems e.g., [32]), information is presented to the user according to his/her preference specifications. This is clearly inappropriate for our problem. Web resource discovery aims to find Web pages relevant to users' requests or interests (e.g., [9] [13][20][21][27]). This approach uses techniques such as link analysis, link topologies, text classification methods to find relevant pages. The pages can also be grouped into authoritative pages, and hubs. However, relevant pages, which are often judged by keywords, are not sufficient for our purpose because we need to further process the contents of the Web pages to discover those salient concepts of the topic and descriptive pages.

A closely related work to ours is question-answering (e.g., [11][17][18][22]). The objective of a question-answering system is to provide direct answers to questions submitted by a user. In this task, many of the questions are about definitions of terms. Early research in this area was ignited by Artificial Intelligence researchers. The aim was to use natural language processing techniques to answer natural language questions. Due to the difficulty of natural language understanding, the techniques were limited to domain-specific expert systems. In the recent years, due to the advances in natural language processing and information retrieval, the interest in question-answering research was re-ignited. A question-answering system typically answers user questions by consulting a repository of documents (see [11] [17][18][22]). [22] also uses the snippet returned from a search engine to help find answers to a question. Our informative page discovery (finding pages containing definitions of concepts) is similar to answering definition questions. We have utilized some of the heuristics from question-answering research for finding such informative pages. However, our whole task is different. We also need to find those sub-topics/salient concepts of the topic from multiple Web pages, and to deal with ambiguity in the search for salient concepts. In terms of definition finding, we also make use of the Web presentation characteristics as clues.

Our work on salient concept discovery from Web documents is related to identifying *collocations* from English text using both statistical and linguistic methods (e.g., [12][31]). Collocations are recurrent combinations of words that co-occur more often than expected by chance. They often represent terminologies or important noun phrases in English text. NPtool [33], a commercial noun phrase detector tool, employs part-of-speech tagging and morphological analysis for this purpose. [6][19] present some heuristic methods for extraction of medical terminologies and their definitions from online documents. [15] also presents an algorithm to find important phrases from documents using a machine learning technique. In our work, we do not require such level of linguistic analysis or learning, which needs a large amount of manually labeled training data. Such techniques also tend to produce too many candidates and most of them may not be important concepts. In the context of the Web, we can exploit the structure of the Web pages to identify candidate phrases. To find a more complete set of key phrases, we study multiple Web pages rather than a single document. Using a data mining technique, the proposed method is able to identify those salient concepts accurately (see the evaluation section). More importantly, the proposed technique integrates the two technologies (definition finding and salient concept discovery on the Web) to perform a novel task, i.e., mining and compiling topic-specific knowledge from Web pages to help the user to perform systematic learning of a topic on the Web.

3. THE PROPOSED TECHNIQUE

The objective of our proposed task is to help the user learn on the Web just like reading a book. Our technique has four iterative steps (Figure 1). The input to the technique is a search phrase *T* representing the topic that one is interested in:

```
Algorithm WebLearn($T$)
1   Submit $T$ to a search engine, which returns a set of relevant pages.
2   The system mines the sub-topics or salient concepts of $T$ using a set $S$ of top ranking pages from the search engine.
3   The system then discovers those informative pages, i.e., those pages containing definitions of the topic and sub-topics (salient concepts) from $S$.
4   The user views the concepts and those informative pages.
    If s/he still wants to know more about the sub-topics then
        for each user-interested sub-topic $T_i$ of $T$ do
            WebLearn($T_i$);
```

Figure 1. The overall algorithm.

In this section, we will discuss these steps. We will first discuss Step 2 and 3 before going to Step 1 and 4, as Step 1 and 4 are fairly straightforward. However, when we deal with the difficult problem of ambiguity of sub-topics, these 2 steps become involved and interesting.

3.1 Sub-Topic or Salient Concept Discovery

The objective of this step is to identify sub-topics and/or salient concepts from an initial set of documents returned by the search engine. It may appear that we will need the natural language understanding ability to find such concepts. This is not so because of the following observation.

Observation: Sub-topics or salient concepts of a topic are the important word phrases. In Web pages, authors usually use emphasizing html tags (e.g., *<h1>,…,<h4> <b>*) to indicate their importance. This serves at least two purposes: (1) it highlights the important concepts to the reader; and (2) it helps to organize the information on the page.

However, it is not sufficient to simply identify and extract those emphasized phrases from the returned pages from the search engine because of the following reasons:

- Web pages are often very "noisy". They typically contain many pieces of unrelated information. Thus, many unrelated text segments may be emphasized.
- Web page authors may emphasize those phrases or even long text segments that are not key concepts of the domain. For example, they tend to emphasize text segments that are related to their work or products, which may not be important sub-topics or key concepts of the domain.

To find those true sub-topics or key concepts of the domain, we need to deal with the above problems. Data mining techniques come to help naturally because they are able to find those frequent occurring word phrases, i.e., those phrases that appear in many pages. Thus, we can eliminate those peculiar ones that appear rarely. Those frequent word phrases are most likely to be the salient concepts of the topic or the domain. This works out very well as we will see in the evaluation section. We now describe the proposed method in detail.

As mentioned earlier, the set of relevant documents is first obtained by using a search engine (in our case Google [7]). After obtaining the set of top ranking pages, sub-topic discovery consists of the following 5 steps:

1. Filter out those "noisy" documents that rarely contain sub-topics or salient concepts: These documents include publication listing pages of researchers, forum discussion pages and pages that do not contain all of the query terms of the topic. The filtering heuristics are based on cue-phrases ('In proceeding', 'journal', 'next message', 'previous message', 'reply to') that appear frequently in these noisy documents. The resulting set of documents serves as the source for sub-topic or salient concept discovery.

2. Identify important phrases in each page: In the Web environment, Web page authors use several HTML markup tags to emphasize important terms or concepts in their documents. Examples of these emphasizing tags include: *<h1>,…,<h4> <b> <strong> <big> <i> <em> <u> <li> <dt>*. Our key concept identification task is facilitated by this fact. However, naive use of these decorated texts turned out to

be harmful. A significant number of them are in fact unconstructive to our task. For this reason, we have identified several rules to determine if a markup text can be safely ignored. The following list summarizes these rules:

- Contains a salutation title (e.g., Mr, Dr, Professor).
- Contains an URL or an email address.
- Contains terms related to a publication (conference, proceedings, journal).
- Contains digits (e.g., EECS2001, WWW10, KDD2002).
- Contains an image between the markup tags.
- Too lengthy (thus unlikely to describe a sub-topic), we use 15 words as the upper limit for a useful emphasized text.

Using these rules, we parse all the pages to extract only quality text segments enclosed by the HTML emphasizing tags listed above. We then perform *stopwords* removal and word *stemming*, which are standard operations in information retrieval. Stopwords are words that occur too frequently in documents and have little informational meanings [30]. Stemming finds the root form of a word by removing its suffix. We use Porter's algorithm [29] for stemming.

3. Mine frequent occurring phrases: Each piece of texts extracted in step 2 is stored in a dataset called a *transaction* set. We then run an association rule miner [24] [1] (which is based on the Apriori algorithm in [1]) to find those frequent itemsets. In our context, an itemset is a set of words (or items) that occur together. Each resulting frequent itemset is a possible sub-topic or salient concept. In our work, we define an itemset as frequent if it appears in more than two documents.

   The Apriori algorithm works in two steps. In the first step, it finds all *frequent itemsets* [2] from a set of *transactions* that satisfy a user-specified *minimum support*. In the second step, it generates rules from the discovered frequent itemsets. For our task, we only need the first step, i.e., finding frequent itemsets, which are candidate sub-topics. In addition, we only need to find frequent itemsets with three words or fewer in this work as we believe that a salient concept contains no more than three words (this restriction can be easily relaxed).

4. Eliminate itemsets that are unlikely to be sub-topics, and determine the sequence of words in a sub-topic: This is a post-processing step. We first remove those unlikely sub-topics (itemsets). We use the following heuristic: If an itemset does not appear alone as an important phrase in any page, it is unlikely to be a main sub-topic and it is thus removed. This heuristic is obvious because if the words in the itemset always appear with some other words together as emphasized texts, it is unlikely to be an important concept. For example, we may find a frequent itemset, {process}. However, since it is always together with other words in an emphasized text, it is removed. Beside this heuristic, we also remove some generic words from the result set (i.e., abstract, introduction, summary, acknowledgement, conclusion, references, projects, research) which appear frequently in writings on the Web.

   Word sequence of a sub-topic is important when we report the results to the user. The Apriori algorithm handles each transaction and frequent itemset as a bag of items (or words) without the notion of sequence. For example, an itemset may be {content mining web}, which should be reported as "web content mining". To overcome this problem efficiently, we lexicographically sort transactions of three words or fewer and store them in a hash table (we assume that a salient concept contains no more than three words). The keys of the hash table are the sorted phrases, while the values are the original word sequences. We can then determine the sequence of words of the frequent itemsets (which are lexicographically sorted by our Apriori algorithm implementation) by hashing each of them into the hash table containing the lexicographically sorted transactions. Note that these two post-processing procedures are performed together.

5. Rank the remaining itemsets: The remaining itemsets (or phrases) are regarded as the sub-topics or salient concepts of the search topic given by user. We rank these concepts based on the number of pages that they occur. This ranking puts those more important concepts on the top of the ranking.

In summary, this step mines those sub-topics or salient concepts of the search topic. Experiment results show that the proposed method works very well. Below, we identify informative pages.

3.2 Definition Finding

This step seeks to identify those pages that contain definitions of the search topic and its sub-topics or salient concepts discovered in the previous step. Precise definition identification requires sound linguistic rules and infallible heuristics. Due to the diversity of the Web and the lack of strict compliant rules, this is unfortunately not a trivial task. However, from our experiments and also previous research (e.g., [11][17][19]), we identified many definition identification patterns that are suitable for Web pages, which involve shallow text processing and conventional definition cues. Instead of using only those emphasized texts, this step goes to the other parts of each page to find definitions.

Noted that texts that will not be displayed by the browsers (e.g., <script>…</script> <!-- *comments* -->) are ignored. In addition, word stemming is applied. However, stopwords and punctuations are kept as they can serve as clues to identify definitions. Note also HTML tags within a text paragraph also need to be removed.

After these preprocessing, the following patterns are applied to identify definitions of concepts:

- {is | are} [*adverb*] {called | known as | defined as} {*concept*}
- {*concept*} {refer(s) to | satisfy(ies)} …
- {*concept*} {is | are} [*determiner*] …
- {*concept*} {is | are} [*adverb*] {being used to | used to | referred to | employed to | defined as | formalized as | described as | concerned with | called} …
- {What is} [*determiner*] {*concept*}?

[1] Association rule mining is stated as follows [1]: Let $I = \{i_1, \ldots, i_n\}$ be a set of items, and D be a set of transactions (the dataset). Each transaction consists of a subset of items in I. An *association rule* is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \varnothing$. The rule $X \rightarrow Y$ holds in D with confidence c if c% of transactions in D that support X also support Y. The rule has support s in D if s% of transactions in D contain $X \cup Y$. The problem of mining association rules is to generate all association rules in D that have support and confidence greater than the user-specified minimum support and minimum confidence.

[2] In [5], an algorithm is also reported to find all maximal itemsets.

- {*concept*} {- | :} {*definition*}
- <dt> {*concept*} <dd> {*definition*}

Legend:	{ }	- compulsory field
	[ ]	- optional field
	adverb	- e.g., usually, normally, generally,…
	determiner	- e.g., the, one, a, an, …
	definition	- definition of a concept

Although some authors use braces (e.g., ( ) < > [ ]) to wrap definitions, they are not used to detect definitions in our work. The reason is that braces are also widely used to wrap examples and illustrations that are often not definitions. Hence, using them to identify definitions reduces the precision significantly. Because of this, we may lose some definition pages. However, this is not a big problem on the Web because the Web often contains many versions of the same information from many authors. Thus, it is not necessary to identify every definition page. Furthermore, the user does not want to see the same definition in many pages. That is, on the Web, recall is often not an issue, but precision is.

Besides using the above patterns to identify definitions, we also rely on HTML structuring clues and hyperlink structures.

1. If a page contains only one header (<h1>, <h2>, …) or one big emphasized text segment at the beginning in the entire document, we assume that the document contains a description/definition of the concept printed in that header/segment. This is reasonable because the page is almost certainly dedicated to the concept.

2. We also discover concept definitions at the second level of the Web hyperlink structure. We make use of the concepts discovered in the previous phase to guide us in this task. Specifically, only hyperlinks with anchor text matching those concepts are taken into consideration. In these second level documents, we determine if they contain definitions of concepts using all the patterns and methods described above. Note that we only look for concept definitions up to the second level of the hyperlink structure, which is sufficient.

Once the defined concepts are identified from each page, we attach the page to each concept, and present to the user. The user can view these informative pages to gain knowledge of the search topic and its salient concepts.

We can also rank the set of Web pages based on the number of concept definitions they contain. As a result of the ranking, pages containing definitions of different key concepts are ranked higher. Users may want to browse such pages first as they are more informative about the topic.

One observation is that in some cases the informative pages returned for each sub-topic or salient concept may not contain sufficient information of the sub-topic. Sometimes, no informative page is found for a particular sub-topic. The reason is that those pages for the main topic are often very general and thus may not contain detailed definitions and descriptions of its lower level concepts. In such cases, we can submit the sub-topic to the search engine and find its sub-subtopics and informative pages recursively (see Figure 1).

3.3 Dealing with Ambiguity

One of the difficult problems in concept mining is the ambiguity of search terms [23]. For example, the term "classification" is so general that it can appear in almost any context, e.g., library classification, product classification, classification in data mining, etc. However, when one is interested in learning a topic, he/she is often only interested in its meaning in a particular context. For example, one may be only interested in "classification" in the data mining context. However, a search engine may not return any page in the right context in its top ranking pages.

In general, search engines are not able to handle this problem directly. However, it can be partially dealt with by adding terms (or words) that can represent the context. Such terms are usually the parent topic of the current topic or sub-topic. For example, if we are interested in "classification" in the context of data mining, we can submit the search query "classification data mining". This method has a shortcoming. That is, the returned Web pages often focus more on the context words, e.g., "data mining", because a context tends to represent a larger topic. This creates a problem because it is harder to find the sub-topics and salient concepts of the more specific topic that we are actually interested in, i.e., "classification" in this case. This is because few returned pages contain in-depth description of "classification". Instead, they may have short descriptions of several sub-topics of data mining (classification is only one such sub-topic). Hence, our method in Section 3.1 may not be able to produce satisfactory results. Instead, it will find many parallel concepts of the search topic (the technique above will rank them high because they tend to be more frequent). For example, for "classification", we may find that the concepts "association rule mining", "clustering", and "sequential rule mining", are ranked high, although they are not sub-topics or concepts of "classification" but of "data mining".

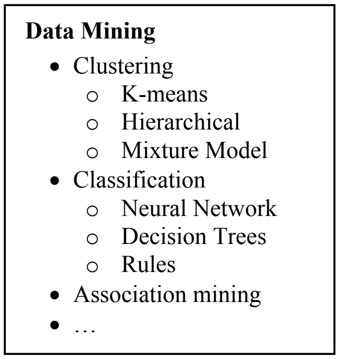
To tackle this problem, we need a more sophisticated technique. As discussed above, we first reduce the ambiguity of a search topic by making use of its parent topic as the context to obtain the initial set of relevant pages from a search engine [3]. These pages, in fact, are quite useful. We have designed the following three methods to help us in discovering sub-topics or concepts from this initial set of documents:

1. Finding salient concepts only in the segment describing the topic (or sub-topic), e.g., "classification": We need to first identify the beginning and end of the segment. To perform this, we rely on several HTML structuring tags (<h1>, <h2>, …, <b>, <strong>, <font size = +…>) as cues. The beginning of the segment is found by locating the first occurrence of the topic enclosed within these tags. The end of the segment is logically the next corresponding tag. If this method fails to identify the topic's segment, we make use of the hyperlink structure. We assume that a hyperlink with anchor text containing the topic points to a page describing it. Salient concepts are then extracted from the topic's segment as discussed in Section 3.1.

2. Identifying those pages that hierarchically organize knowledge of the parent topic: In some cases, a Web page may already have a well-organized sub-topic hierarchy, just like the table of contents in a typical book. For example, for the topic "data mining", we may find pages such as the example in Figure 2. Such Web pages provide important clues for finding sub-topics. They are also valuable pages for a

[3] Currently, we are still unable to handle the case when the first (search) topic from the user's initial query is also ambiguous. We plan to exploit query expansion techniques to deal with the problem in the future.

learner to focus on. To identify such pages, we can parse the HTML nested list items (e.g., <li>) structure by simply building a tree structure. The branches with root containing the search topic are regarded as the sub-topics.



Data Mining

- Clustering
  - K-means
  - Hierarchical
  - Mixture Model
- Classification
  - Neural Network
  - Decision Trees
  - Rules
- Association mining
- …

Figure 2. Example of a well-organized topic hierarchy

After such a page is discovered, we need further evidence to show that it is a right page. For example, the hierarchy containing "classification" may be a product classification page. We confirm whether it is a correct page by finding if the hierarchy also contains at least another sub-topic of the parent topic. For example, in the case of "classification", we need to find whether the hierarchy also describes another parallel concept of "classification" in data mining. In Figure 2, we can see that it does, i.e., "clustering". With both pieces of evidence, we are reasonably confident that the page is a right page. We then extract those sub-concepts between "classification" and the next concept in parallel (or the page end), e.g., "association mining". Note that sub-topics discovered using this technique must also be no longer than three words. Otherwise, they are unlikely to be a concept. Note also that this method assumes that we already know the sub-concepts of the parent concept (e.g., "data mining"). Thus, it is only applicable to situations where we want to find salient concepts of the sub-topics of a big topic.

3. Finding salient concepts enclosed within braces illustrating examples: Web page authors often use braces "( )" to enclose important concepts such as in the following example:

   > *There are many clustering approaches (e.g., hierarchical, partitioning, k-means, k-medoids), and we add that efficiency is important if the clusters contain many points.*

   Based on this fact, we can discover sub-topics by identifying sentences containing such information. These sentences must first contain the search topic followed by an optional cue-phrase, i.e., "approaches", "techniques", and "algorithms". There must also be more than one example between the braces. We detect this by identifying multiple commas. 'e.g.', 'such as', 'for example' and 'including' indicate what follow are examples. Additionally, each example within the braces must not have more than 3 words.

In summary, the above technique is very effective, as we will see in the evaluation section. By no means, however, do we claim that the ambiguity problem is completely solved using this approach. If there are no suitable words to describe the context of the topic that one is interested in, the above approach will not apply. Further research is still needed.

It is also important to note that methods 1 and 3 above may also be used for finding salient concepts in Section 3.1. However, we found that they are not necessary if we do not need context words in the search. As for method 2, it is unreliable to use it if we do not know the parallel concepts of the search topic (e.g., "classification") because it is hard to decide whether a hierarchy like the one in Figure 2 is a right page. It may be a product classification page or library classification page. However, if the hierarchy also contains some parallel concepts of "classification", we can be sure that it is very likely to be a right page.

Finally, we answer the following question: how does one know that there is no need to go down further to find salient concepts of a topic or a sub-topic (line 4 in Figure 1)? We should stop when we find that many salient concepts are actually parallel concepts of the search topic or sub-topic. In practice, the user can also detect when to stop easily.

3.4 Mutual Reinforcement

Although the above techniques are effective in performing their tasks, we can do better in many cases. This section presents a mutual reinforcement method to improve the proposed technique further. This method applies to situations where we have already found the sub-topics of a topic, and we want to find the salient concepts of the sub-topics of the topic, i.e., to go down further.

A naïve method is to apply the methods presented so far from Section 3.1 till now on each sub-topic individually to find its salient sub-subconcepts. This may be sufficient. However, in many cases, the sub-topics can help each other.

It is often the case that in the pages returned by searching one sub-topic S_1 we can find some important information about another sub-topic S_2 as the sub-topics are related. However, such pages may not appear at the top of the search results when we search for the sub-topic S_2 due to the idiosyncrasy of the ranking algorithm used by the search engine. For example, when we search for "classification data mining", we may find that some pages also contain useful information about "clustering", such as the page in Figure 2. This page, however, may not appear as a top-ranking page when searching for clustering. In the same way, the search for "clustering data mining" ("clustering" is also ambiguous) may find a page that is very useful for classification.

We implemented this technique in two steps: (1) submit each sub-topic individually to the search engine; (2) combine the top ranking pages from each search into one set; and apply the proposed techniques to the whole set to look for all sub-topics.

4. SYSTEM ARCHITECTURE

We have implemented a system (called WebLearn) based on the proposed framework. The entire system is coded in PERL and can be executed efficiently in both Microsoft Windows and Unix Environment. Figure 3 shows the overall system architecture. It consists of 5 main components.

1. A search engine: This is a standard Web search engine.
2. A crawler: It crawls the World Wide Web to download those top ranking pages returned by the search engine. It stores the pages in "Web Page Depository".
3. A salient concept miner: It uses the techniques presented in Sections 3.1-3.4 to search the pages stored in "Web Page Depository" to identify and extract those sub-topics or salient

concepts.

4. A definition finder: It uses the technique presented in Section 3.2 to search through the pages stored in “Web Page Depository” to find those informative pages containing definitions of the topics and sub-topics.
5. A user interface: It enables the user to interact with the system.

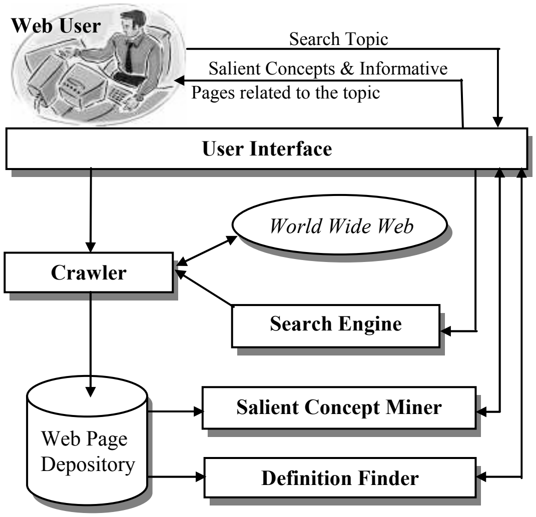


Figure 3. System architecture of WebLearn.

5. EXPERIMENTAL STUDY

This section evaluates the proposed technique and our system WebLearn. We use the Google [7] search engine to obtain the initial set of relevant documents for mining. The size of this set of documents is limited to the first hundred results (100) returned by Google. Experiments suggest that using a larger set of documents does not help and sometimes, it may be harmful. The reason for this is that additional documents are in general less reputable and thus less informative than the first hundred.

Table 1 shows the sub-topics and/or salient concepts discovered for 28 search topics. These topics (or queries) were provided by two graduate students. They were asked to select some well known and diverse topics in computer science so that our results can also be evaluated easily by other readers. In building our system, we used three topics to test our system, namely, artificial intelligence, data mining, and Web mining (which are also included in Table 1).

In each box, the first line gives the topic (the exact words are used in the Google search). For each topic, we listed only ten top-ranking concepts due to space limitations. From Table 1, we can see that our technique is able to effectively find major sub-topics of each query topic. Note that for too specific topics, only definition finding is meaningful.

In Table 2, we compare the precision (in %) of our definition-finding task with the Google [7] search engine and AskJeeves [4], the Web’s premier Question-Answering System (we are unable to compare with the system in [22] as it is not available on the Web). Recall is not evaluated since computing recall on the Web remains an intricate task and it is often not an issue because many results are returned in the Web context. We compare the first 10 pages of our results with the first 10 pages returned by Google and AskJeeves. To do a fair comparison, we also look for definitions in the second level of the search results returned by Google and AskJeeves. In addition, if our system returns less than 10 pages with definitions of a particular topic, we will compare with that corresponding number of Google and AskJeeves’s results. Column 1 of Table 2 shows the 28 search topics (or queries) in Table 1, while column 2 displays the precision of definition finding of our system for each topic. Columns 3 and 4 show Google’s and AskJeeves’s precision results respectively. The final row gives the average precision of each column. From Table 2, we can see that on average the precision of our system, WebLearn, is much better than those of Google and AskJeeves. Out of the 28 search topics, WebLearn produces better results than both systems on 26 topics. Only for topic 11, Google has a slightly better precision. On topic 17, no system is able to find any definition page. All the pages are inspected by two independent judges. We did not use more judges as definitions are fairly easy to check and are not very subjective.

Table 3 presents our results for ambiguity handling by applying the methods in Section 3.3 and also 3.4 (which is also useful for non-ambiguous topics). Note that due to space limitations, only top ranking (more frequent) sub-topics are shown. Column 1 lists two ambiguous topics of “data mining” (i.e., “classification”, “clustering”) and two ambiguous topics of “time series” (i.e., “smoothing”, “models”). Column 2 lists the sub-topics or salient concepts identified using the original technique in Section 3.1. Due to the ambiguity of these search topics, the results are clearly unsatisfactory. Column 3 gives the sub-topics or salient concepts discovered using the respective parent-topics as context terms, in addition to employing the technique in Section 3.1. Using this approach, we discover mostly those parallel concepts of each search topic. For example, “data visualization”, “association rules” and “clustering algorithms” are all parallel concepts of “classification”, i.e., they are sub-topics of “data mining”. However, when we apply the ambiguity handling methods in Section 3.3, we are able to achieve much better results (Column 4). We can see that almost all of them are indeed sub-topics/salient concepts of “classification” or “clustering” in the context of “data mining”. The same applies to the search topics under “time series”. In the last column, we show the discovered sub-topics when the ‘mutual reinforcement’ technique discussed in Section 3.4, is also employed. The results are further enhanced to some extent. For example, for “clustering”, we found “Agglomerative”, which is one of the important clustering techniques. For “models” under “time series”, we found additional sub-topics such as “linear” and “additive” models. Note that in all the experiments, we only find one page on data mining that contains a sub-topic hierarchy (see Section 3.3).

Execution time: We now discuss the running efficiency of our system. We use a modest machine (Intel Pentium III 866 MHz, 128MB memory, single processor) for all our experiments. The system is implemented in the PERL language. We also implemented a caching utility on our system to eliminate repeated crawling of Web pages. On average, each of the 28 queries in Table 1 took 2 min 31 sec, which include reading in each page, parsing, association mining and finding definitions. Improving the efficiency of crawling and parsing has not been the main focus of this work. With further optimization and by using a faster machine and a more efficient language (e.g., C/C++) for parsing of Web pages, the running speed can be significantly improved.

Table 1. Experiment Results of Sub-topic/Salient Concept Discovery.

Artificial Intelligence: Machine learning Robotics Philosophy Neural networks Expert systems Games Artificial life Vision Natural language processing Connectionism	**Data Mining:** Clustering Classification Data Warehouses Databases Knowledge Discovery Web Mining Information Discovery Association Rules Machine Learning Sequential Patterns	**Web Mining:** Web Usage Mining Web Content Mining Data Mining Webminers Text Mining Personalization Information Extraction Semantic Web Mining XML Mining Web Data	**Machine Learning:** Neural Networks Artificial Intelligence Inductive Logic Programming Data Mining Computational Learning Theory Information Retrieval Games Reinforcement Learning Decision Trees Genetic Algorithms
Computer Vision: Motion Object Recognition Image Processing Vision Systems Computer Graphics Computer Vision Syndrome Image Formation Perceptual Grouping Artificial Intelligence Image Understanding	**Relational Calculus:** Relational Algebra Tuple Relational Calculus Domain Relational Calculus SQL Index Extended Relational Calculus Integrity Constraints Relational Model Microsoft Access Predicate Logic	**Linear Algebra:** Determinants Linear Transform Vectors Spaces Matrix Algebra Matrices Mathematical Eigenvectors Similarity Echelon Form Null Space	**Neural Network:** Back Propagating Training Perceptron Neural Computation Genetic Algorithms Self Organizing Maps Neural nets Nets SOM Multi Layer Perceptron
Fuzzy Logic: Fuzzy Sets Logic Fuzzy Controllers Fuzzy Logic Controllers Neural Networks Artificial Intelligence Fuzzy Numbers Fuzzy Set Theory Fuzzy Systems Fuzzy Logic Applications	**Time Series:** Exponential Smoothing Series Periodicity Frequency Forecasting Time Series Data Trends Smoothing Moving Averages Models	**Query Languages:** XML query languages Indexing Tuple Relational Calculus Relational Algebra Microsoft Access Query Optimization Relational Calculus Domain Relational Calculus Data Model Structure Preserving	**Question Answering:** Question Answering Systems Computational Linguistic Semantic Relation Tuples Answer Extraction Answer Selection Search Engines Wall Street Journal Question Classification Query Expansion Question Parsing
Bioinformatics: Databases Proteins Genetics Computational Biology Bioinformatic Group Embnet Informatics BMC Bioinformatics Cambridge Heathtech Institute Human Genome	**Database Design:** Relational database design Tables Programming Views Entity relationship diagrams Development Adminstration Relational databases Relationships Data modeling	**Genetic Algorithm:** Mutation Crossover Selection Genetic Programming Fitness Function Fortran Genetic Algorithm Algorithms Gene Evolutionary Computation Optimization	**Information Retrieval:** Digital Libraries Modern Information Retrieval Indexing Images Relevance Feedback Internet Modeling Search Engines Information Processing Machine Learning
Parallel Computing: PVM MPI Beowulf Networks Cluster Computing Distributed Computing Parallel Computing Works Parallel Programming Computer Engineering Parallel Machines	**Computer Architecture:** Parallel Computer Architecture Architectures Instruction Sets Workload Characterization Operating Systems Cache Memory Multi Threaded	**Linear Regression:** Slopes Intercept Assumptions Residuals Multiple Linear Regression Simple Linear Regression Probability Selecting Test Multiple Regression	**Computer Security:** Hackers Firewalls Privacy Advisories Coast Cryptography Information Warfare Exploits Encryption WWW Security
Natural Language Processing: Information Retrieval Natural Language NLP Machine Translation Information Extraction Computational Linguistic Language Engineering Noun Phrase Speech Recognition Corpus Linguistic	**Computer Graphics:** Animations Rendering Multimedia Virtual Reality Computer Science Departments OpenGL Computer Animation Computational Visualization Graphics Programming Clip Art	**Software Engineering:** Engineering Requirements Engineering Testing Case Tools Problem Sets Project Management IEEE Software Formal Methods Nie Logiciel Software Engineering Standards	**Perceptual Grouping:** Computational Vision Perception Segmentation Texture Perceptual Organization Good Continuation Aerial Images Vision Research Neural Networks Gestalt Psychology
Firewall: Features Proxy Servers Security Logging Policies Port Filtering Packet Filtering Linux Firewall Personal Firewall	**Automata Theory:** Languages Push Down Automata Finite Automata Regular Expressions Turing Machines Cellular Automata Context Free Grammars Theory Grammars Normal Forms	**Web Caching:** Squid Proxy caching Adaptive Web Caching Transparent caching Multicast World Wide Web Servers Cache Hierarchies Web Caching Architecture Web Servers	**Constraint Satisfaction:** Constraint Satisfaction Problem Variables Domains Satisfiability Artificial Intelligence Arc Consistency CSPs Scheduling Cycle Cutset Evolutionary Algorithms

Table 2. Precisions of definition finding.

Search Topic	WebLearn	Google	AskJeeves
1. Artificial Intelligence	50.00	0.00	0.00
2. Data Mining	70.00	30.00	10.00
3. Web Mining	75.00	37.50	50.00
4. Machine Learning	77.78	22.22	11.11
5. Computer Vision	33.33	0.00	0.00
6. Relational Calculus	83.33	33.33	50.00
7. Linear Algebra	40.00	00.00	0.00
8. Neural Network	80.00	30.00	20.00
9. Fuzzy Logic	90.00	20.00	40.00
10. Time Series	50.00	0.00	0.00
11. Query Languages	20.00	30.00	20.00
12. Question Answering	75.00	0.00	0.00
13. Bioinformatics	60.00	10.00	0.00
14. Database Design	83.33	33.33	0.00
15. Genetic Algorithm	100.00	30.00	0.00
16. Information Retrieval	50.00	0.00	0.00
17. Parallel Computing	0.00	0.00	0.00
18. Computer Architecture	66.67	0.00	0.00
19. Linear Regression	60.00	50.00	50.00
20. Computer Security	33.33	0.00	0.00
21. Natural Language Processing	100.00	0.00	33.33
22. Computer Graphics	75.00	25.00	25.00
23. Software Engineering	16.67	0.00	0.00
24. Perceptual Grouping	66.67	50.00	33.33
25. Firewall	60.00	40.00	30.00
26. Automata Theory	33.33	0.00	25.00
27. Web Caching	75.00	25.00	25.00
28. Constraint Satisfaction	90.00	50.00	50.00
Average:	**61.23**	**18.44**	**16.88**

6. CONCLUSION

This paper proposed a novel task and also a set of initial techniques for finding and compiling topic specific knowledge (concepts and definitions) on the Web. The proposed techniques aim at helping Web users to learn an unfamiliar topic in-depth and systematically. We have also built a prototype system that implements the proposed techniques. Given a topic, the system first discovers salient concepts of the topic from the documents returned by the search engine. It then identifies those informative pages containing definitions of the search topic and its salient concepts.

Due to the convenience of the Web along with its richness and diversity of information sources, more and more people are using it for serious learning. It is important that effective and efficient systems be built to discover and to organize knowledge on the Web, in a way similar to a traditional book, to assist learning. This is the long-term objective of our project. We believe that this work represents an important step toward this direction. In our future work, we will also study how hyperlinks and meta-data can be used in the process to produce even better techniques. We also plan to study how the proposed technique can be implemented in a search engine environment so that a search engine can provide the same service with better efficiency.

7. REFERENCES

[1] Agrawal, R. & Srikant, R. “Fast algorithm for mining association rules.” VLDB-94, 1994.

[2] Anderson, C. and Horvitz, E. “Web Montage: A Dynamic Personalized Start Page.” WWW-02, 2002.

[3] Ashish, N. & Knoblock, C. “Wrapper generation for semi-structured Internet sources.” SIGMOD Record, 26(4), 1997.

[4] AskJeeves, Inc., AskJeeves Question-Answering Search Engine, http://www.ask.com.

[5] Bayardo, R. “Efficiently Mining Long Patterns from Databases.” SIGMOD-98. 1998.

[6] Bennett, N.A., He, Q., Powell, K., Schatz, B.R. “Extracting noun phrases for all of MEDLINE.” In Proc. American Medical Informatics Assoc., 1999.

[7] Brin, S. & Page, L. “The anatomy of a large-scale hypertextual web search engine.” WWW7, 1998.

[8] Ceri, S., Comai, S., Damiani, E., Fraternali, P., & Tranca, L. “Complex queries in XML-GL.” In SAC (2) 2000:888-893.

[9] Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S, Gibson, D. & Kleinberg, J. “Automatic resource compilation by analyzing hyperlink structure and associated text.” WWW7, 1998.

[10] Cohen, W., Fan, W. “Learning page-independent heuristics for extracting data from Web pages.” WWW8, 1999.

[11] Cooper, R.J. & Rüger, S. M. “A simple question answering system.” In Proc. of TREC 9, 2000.

[12] Daille, B. “Study and implementation of combined techniques for automatic extraction of terminology.” In The Balancing Act: Combining Symbolic and Statistical Approaches to Language. The MIT Press, 1996.

[13] Dean, J. & Henzinger, M.R. “Finding related pages in the World Wide Web.” WWW8, 1999.

[14] Feldman, R., Liberzon, Y., Rosenfeld, B, Schler, J. & Stoppi, J. “A framework for specifying explicit bias for revision of approximate information extraction rules.” KDD-00, 2000.

[15] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. & Nevill-Mainning, C.G. “Domain-specific keyphrase extraction.” In IJCAI-99, 1999.

[16] Guan, T. & Wong, K.F. “KPS – a Web information mining algorithm.” WWW8, 1999.

[17] Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V. & Morarescu, P. “FALCON: Boosting knowledge for answer engines.” In Proc. of TREC-9, 2000.

[18] Katz, B. “From sentence parsing to information access on the WWW.” In AAAI Spring Symposium on Natural Language Processing for the WWW, 1997.
http://www.ai.mit.edu/projects/infolab/ailab.html

[19] Klavans, J. L. & Muresan, S. “DEFINDER: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text.” In proc. of American Medical Informatics Assoc., 2000.

[20] Kleinberg, J. “Authoritative Sources in a Hyperlinked

Table 3. Results of ambiguity handling.

Search Topic	Original methods in Section 3.1	With context terms (Methods in Section 3.1)	Ambiguity handling (Methods in Section 3.3)	Mutual reinforcement (Methods in Section 3.3 & 3.4)
Data Mining				
Classification	Incertae Sedis Content Critical Related links Data mining Classification society Technological change Nlm classification Related topics Background information	Data mining Data visualization Neural networks Modeling Decision trees Association rules OLAP Time series Knowledge discovery Clustering algorithms Visualization	Neural networks Trees Naive bayes Decision trees K nearest neighbor Regression Neural net Sliq algorithm Parallel algorithms Classification rule learning ID3 algorithm C4.5 algorithm Probabilistic models	Neural networks Trees Naive bayes Decision trees K nearest neighbor Regression Neural net Sliq algorithm Parallel algorithms Classification rule learning ID3 algorithm C4.5 algorithm Probabilistic models Controlling model complexity
Clustering	Features Methods Clustering services Communications Beowulf Fail over Databases Similarity High availability Server clustering Psychological review	Data mining Classification Neural networks Decision trees Models Spatial data mining Web mining Machine learning Time series Statistics Databases	Hierarchical K means Density based Partitioning K medoids Distance based methods Mixture models Graphical techniques Intelligent miner	Hierarchical K means Density based Partitioning K medoids Distance based methods Mixture models Graphical techniques Intelligent miner Agglomerative Graph based algorithms
Time Series				
Smoothing	Simple exponential smoothing Moving average Time series Median filter Data smoothing Gaussian smoothing Font smoothing Smoothing parameters Smooth edges	Time series analysis Arma Moving averages Smoothness Spectral analysis Auto correlation Modeling Seasonal decomposition Trend Exponential smoothing	Simple exponential Moving averages Double exponential Trend Triple exponential Simple exponential smoothing Exponential smoothing Seasonal decomposition Single exponential Multiple regression	Simple exponential Moving averages Double exponential Trend Triple exponential Simple exponential smoothing Exponential smoothing Seasonal decomposition Single exponential Multiple regression Partial autocorrelations
Models	Featured models Rho models Female models Updated daily Glamour models New faces Scale modeller Movie reviews Internet modeler Model available Models needed Realspace models	Time series analysis Multivariate analysis Forecasting Algorithms Graphics Smoothness Programming Statistical inference Time series modeling Simulation Exponential smoothing Seasonal decomposition	Nonlinear Arma Garch Cycle Arima Stationarity Local linear trend Combined gmm estimators Multinomial logit Box jenkins approach Descriptive statistics	Nonlinear Arma Garch Cycle Arima Stationarity Local linear trend Combined gmm estimators Multinomial logit Box jenkins approach Descriptive statistics Linear Additive

Environment." Proc. of ACM-SIAM Symposium on Discrete Algorithms, 1998.

[21] Kumar, S., Raghavan, P., Rajagopalan, S., Tomkins, A. "Extracting large-scale knowledge bases from the Web." VLDB-99, 1999.

[22] Kwok, C., Etzioni, O. & Weld, D.S. "Scaling question answering to the Web." WWW10, 2001.

[23] Lawrence, S. "Context in Web Search." IEEE Data Engineering Bulletin 23(3): 25-32, 2000.

[24] Liu, B., Hsu, W. Ma, Y., "Integrating classification and association rule mining." KDD-98, 1998.

[25] Maarek, Y. and Shaul, I "Automatically organizing bookmarks per contents." WWW5, 1996.

[26] Mendelzon, A., Mihaila, G. & Milo, T. "Querying the World Wide Web." Journal of Digital Libraries 1(1): 68-88, 1997.

[27] Ngu, D.S.W. and Wu, X. "SiteHelper: A localized agent that helps incremental exploration of the World Wide Web." WWW6, 1997.

[28] Page, L., Brin, S., Motwani, R. & Winograd, T. "The PageRank citation ranking: Bringing order to the Web." In Stanford CS Technical Report, 1998.

[29] Porter, M.F. "An algorithm for suffix stripping." Program 14(3):130-137, 1980.
http://www.tartarus.org/~martin/PorterStemmer/

[30] Salton, G. & McGill, M.J. Introduction to modern information retrieval. McGraw-Hill, 1983.

[31] Smadja, F. "Retrieving collocations from text: Xtract" In Using Large Corpora. London: MIT Press pp143-177, 1994.

[32] Underwood, G. Maglio, P. & Barrett, R. "User-centered push for timely information delivery." WWW7, 1998.

[33] Voutilainen, A. "NPtool: A detector of English noun phrase." In Proc. of Workshop on Very Large Corpora, 1993.