Knowledge Encapsulation for Focused Search from Pervasive Devices

Yariv Aridor, David Carmel,
Yoëlle S. Maarek, Aya Soffer
IBM Research Lab in Haifa
Matam, Haifa 31905, Israel
{aridor,carmel,yoelle,ayas}@il.ibm.com

Ronny Lempel
Department of Computer Science, Technion
Haifa 32000, Israel
rlempel@cs.technion.ac.il

ABSTRACT

Mobile knowledge seekers often need access to information on the Web during a meeting or on the road, while away from their desktop. A common practice today is to use pervasive devices such as Personal Digital Assistants or mobiles phones. However, these devices have inherent constraints (e.g., slow communication, form factor) which often make information discovery tasks impractical.

In this paper, we present a new focused-search approach specifically oriented for the mode of work and the constraints dictated by pervasive devices. It combines focused search within specific topics with encapsulation of topic-specific information in a persistent repository. One key characteristic of these persistent repositories is that their footprint is small enough to fit on local devices, and yet they are rich enough to support many information discovery tasks in disconnected mode. More specifically, we suggest a representation for topic-specific information based on "knowledge-agent bases" that comprise all the information necessary to access information about a topic (under the form of key concepts and key Web pages) and assist in the full search process from query formulation assistance to result scanning on the device itself. The key contribution of our work is the coupling of focused search with encapsulated knowledge representation making information discovery from pervasive devices practical as well as efficient. We describe our model in detail and demonstrate its aspects through sample scenarios.

Keywords

Focused search; Pervasive devices; Disconnected search; Knowledge agents

1. INTRODUCTION

Pervasive devices, such as mobile phones and Personal Digital Assistants (PDAs), currently provide much more functionality than they were originally designed for. PDAs are no longer simple organizer/calendar tools. Hundreds of applications are available on the PalmOS platform alone (from ebooks to Web browsing), both in connected and disconnected modes (see palm.net webclipping applications [14]). This trend can be observed in the mobile phone business as well, with WAP phones offering a wide variety of services ranging from flight information to remote banking.

As these devices developed additional functionality, they have evolved into general purpose information appliances in the form of very thin clients. Information access requires adequate information discovery abilities, including mechanisms for searching and browsing information. The nature of pervasive devices, however, imposes new constraints on classic search and browse paradigms prevalent in the connected desktop world. These constraints are mostly due to:

1. The form factor of pervasive devices. Entering information is extremely cumbersome on pervasive devices. Whether the input device is a stylus supporting graffiti or handwriting recognition, a "soft keyboard" (i.e., a virtual keyboard shown on the display), or a telephone keypad, the rate and accuracy of the input are quite limited. Output is also limited due small screen sizes.

2. The communication mode of pervasive devices. Communication between pervasive devices and external resources varies from sporadic synchronization with a personal computer (via a cradle or modem) to full connectivity via external wireless modems, or built-in connectivity (Palm VII [14]), and RF (RIM handheld devices [16]). The communication mode directly influences the timing of discovery tasks. In the first case, most discovery tasks need to be performed in batch (during synchronization), while in the second case we may wish to minimize the amount of information that is transferred over the wire in order to improve response time, cut costs, conserve battery consumption, reduce radiation exposure and more.

The form-factor-related constraints have a direct impact on the success of any PDA application. Indeed, if the mode of interaction is awkward or too time consuming, the application will most likely not be adopted by users. In the context of WWW information discovery, two types of interaction are critical: navigation (for browsing and exploring) and text input (for searching). Many user studies have been conducted in order to evaluate various methods for information navigation and for text input given various form factor constraints. As discussed in [4], many browsing paradigms that we take for granted in the desktop world have to be reconsidered in the context of small devices. Consequently, content is often pre-packaged in device-dependent formats which simplify the navigation task. Examples include WML decks for WAP phones [21], and HDML pages for Palm VII. Ideally, content should be automatically or semi-automatically translated from a common format (e.g., XML) into device-specific formats, and in fact many transcoding products [18] tackle exactly this problem. However, this ideal solution is not realizable at this point since content is currently not provided in one common format, and since transcoding has turned out to be an elusive task.

Since an all-in-one solution does not seem near, it is necessary to devise interim application-specific and device-specific solutions. It is suggested in [8], for example, that navigation can be enhanced by the definition of short-cut keys that facilitate faster browsing through WML decks in the context of a business card application on WAP phones. A promising direction has also been explored in the Power Browser system [4] that reorganizes the content of individual Web sites so as to decouple the navigation and viewing phases. This method was designed for local site search.

In terms of text input, several approaches have been proposed to enhance the user's experience for various input devices. Tegic T9 [19] uses a disambiguation mechanism which enables text input on phone keypads with much fewer keystrokes than is usually necessary. Masui [12] provides both word-level completion (via menus and dynamic approximate string matching) and phrase-level completion. The Power Browser [4] supports site-specific word completion.

The second class of constraints we consider pertains to the communication mode of pervasive devices, especially in terms of bandwidth limitations and response time requirements. Even if the interaction process is greatly enhanced by the techniques described above, a typical search process still requires numerous transactions between the wireless device and the content-storing servers. In [4] for instance, where users can search a Web site via a wireless PDA, at least six interactions are necessary before PDA formatted search results are displayed. Even with reasonable bandwidth, each such transaction is noticeable by the user who will consequently wait a few seconds for meaningful results.

A common solution geared to improve response time in many computer applications is to cache some information where it is required and thus provide quasi immediate feedback without having to access the original storage location. In the context of pervasive devices, we would like to store/cache some information on the local device, thus eliminating the need to access remote servers during some stages of the information discovery task. AvantGo [2], for example, enables caching a pre-selected set of Web pages on pervasive devices. However, caching data for general purpose Web information discovery tasks would require an insurmountable amount of local storage space.

Focused Search

The constraints inherent to pervasive devices such as PDAs presented above, suggest that a promising direction for making search practical on pervasive devices is using a focused topic-specific search model, that works mostly in disconnected mode, with periodic downloading of data to the device. Focused search addresses the above issues as follows:

- The complexity of navigation on pervasive devices can be further reduced if, rather than focusing on Web sites as in [4], we focus the browsing on specific topics. In particular, topic discovery and browsing is possible by revealing various aspects of the topic-specific knowledge to the user such as top phrases and top Web pages. A topic-specific approach also seems more practical as most discovery tasks involve compiling information from various resources on a specific topic, rather than exploring an individual Web site.

- Working at the topic level allows us to further enhance input completion techniques. We can provide accurate and useful topic-specific query formulation assistance, which includes word completion as well as phrase completion and expansion.

- Addressing the bandwidth limitations of pervasive devices and the response time constraints which applications must meet, we note that limiting ourselves to topic-specific discovery tasks greatly reduces the required amount of cached data, and thus caching based techniques become feasible.

Focused search and crawling has been proposed in the past, mostly with the motivation of assisting novice Web surfers and/or improving the quality (precision) of results. Some systems and services predefine the topics of interest (a.k.a. domains) themselves (see for instance various directory services such as Yahoo! [22] and Google [7]), while others allow users to define topics of interest (e.g., Fetuccino [3], Focused Crawler [6], WTMS [13], and knowledge agents [1]).

Since the amount of memory available on pervasive devices is very limited, it is crucial that the information stored locally be highly representative of the topic, yet concise. While the focused search techniques mentioned above can be used to collect a representative set of Web pages for a particular topic, they are insufficient in terms of encapsulating the knowledge required to support all of the information discovery tasks on the pervasive device.

Our approach

In this paper, we describe a focused-search approach specifically designed for the mode of work and the constraints dictated by pervasive devices. Topics are represented and stored in a persistent repository with a very small footprint which can be downloaded to small devices. More specifically, we suggest representing topic-specific information via "knowledge-agent bases" (KABs) that encapsulate information required to assist users in their discovery process from pervasive devices. The idea is to allow users to predefine a topic of interest, and then capture a very small but representative piece of the Web for this topic, storing characteristic information regarding it on the PDA. The information we store includes a topic-specific lexicon, a small number (~ 100) of the most authoritative Web pages and hubs of the domain (core pages) whose text is stored in its entirety, and a relatively larger number (on the order of thousands) of second-tier Web pages (outlinks of the core pages) for which we store only the URL and anchor text of the outlink. An index which supports search on the text of the core pages as well as on the anchor text of the referenced pages is also stored on the device.

KAB's are built automatically using the Knowledge Agent technology described in [1]. Users need only to supply several (4-6) queries which define the topic, along with an optional set of seed URLs which they consider to represent the topic at hand. We elaborate on the knowledge acquisition process in Section 2.2.

Combining focused search with encapsulated knowledge representation holds the promise of making information discovery from pervasive devices practical as well as efficient. The advantages of this approach are numerous:

1. Response time for many initial discovery tasks is negligible, as they can be performed locally on the device, in fully disconnected mode. Examples include such tasks as query refinement, URL browsing, and searches within a representative set of Web pages.

2. Enhanced query input is supported via topic-specific word and phrase completion.

3. KAB's can be easily exchanged with colleagues since their footprint is rather small.

4. The domains are user defined and can be of any granularity as they are created via several seed URLs and a few sample queries.

5. Topic discovery and browsing is supported by revealing various aspects of the knowledge base such as the top phrases in the lexicon and the list of core pages.

6. Performing tasks in a disconnected mode is in itself a useful feature to users who are unable to connect to a server since they are in remote location not well covered by their wireless service, or in the middle of a meeting, or whether they wish to reduce battery consumption and radiation exposure.

The rest of this paper is organized as follows. Section 2 presents the topic specific knowledge representation and acquisition methods we use. Section 3 describes the process of information discovery in disconnected mode including topic browsing, query formulation, and disconnected search. Section 4 consists of some sample scenarios and examples using knowledge encapsulation for focused search on a Palm device. Section 5 concludes by reiterating the key contributions of this work and pointing to future directions. Related work is discussed as relevant throughout the paper.

2. TOPIC SPECIFIC KNOWLEDGE REPRESENTATION AND ACQUISITION

Knowledge agents is a methodology for focused information discovery that the authors of this paper introduced in [1]. The goal of the knowledge agent approach is to gain persistent knowledge on a given topic. Knowledge agents are created by users, and undergo a training phase during which they acquire automatically expertise on a given topic. This knowledge is subsequently leveraged to facilitate both the search task (query formulation assistance, disconnected search, result ranking) and the browsing (by providing a set of pertinent pages and key concepts) on matters pertaining to the agent's domain of expertise.

The motivation for using a knowledge agent approach in our context is its compact representation, or encapsulation, of topic-specific knowledge. The compactness of the representation allows its deployment in the limited resource environment of pervasive devices, while the topic specificity of the represented knowledge answers the form-factor and communication model constraints of these devices.

The key adaptation of the knowledge agent approach to pervasive information discovery is the physical detachment between the knowledge acquisition phase and the knowledge application phase. Knowledge acquisition is done on a desktop server, with Internet access. After the topic-specific knowledge is acquired (and represented in a KAB), it is downloaded to the pervasive (and often disconnected) device, and serves to conduct as many information discovery tasks as possible either in fully disconnected mode or with minimal access to remote servers.

Figure 1 captures the server/device interaction. The knowledge acquisition phase can be initiated either from a connected desktop or from a PDA connected through a cradle or a wireless modem. Once the KAB is ready, it can be downloaded to the device again through a cradle or wireless modem. The user will only need to reconnect to the server to obtain updates or create new agents. Note that small KABs can also be obtained directly from other users, in a peer-to-peer communication exchange through direct beaming (a small KAB of about 400K takes about two minutes to exchange). Larger KABs can still be exchanged between users (an average KAB of about 1MB can take a few minutes). However, since such a long beaming operation might be inconvenient (not to mention battery consumption problems), it is preferable in the case of larger KABs to beam only a reference to the KAB on the server, and let the receiving party download it directly from the server.

2.1 Knowledge Representation

In order to design a topic-specific, Web oriented knowledge acquisition process, we must first answer the following questions:

1. What is a topic on the Web?

2. How will the creator define the topic in question to the Knowledge Agent?

3. What should the agent learn about the topic, in order to assist future users' information retrieval tasks?

We answer the first question in the broadest manner possible, by defining a topic as a set of thematically related URLs. This definition poses no restrictions on the granularity of topics, and indeed users may create agents for topics of any breadth. With this definition, two outright manners for defining a specific topic t to an agent come to mind:

1. Specifying some members of the set of URLs. The creator may explicitly enter URLs which pertain to the topic t of the agent. While this form of input is allowed at any point in the training phase of the agent, the typical scenario is for these URLs to be given at the time of the agent's initialization. The URLs themselves should be deemed by the creator as being valuable Web resources on t, and may originate, for example, from the creator's personal bookmarks files.

2. Specifying the thematic relationship between the pages of the set. This is done by providing the agent with a

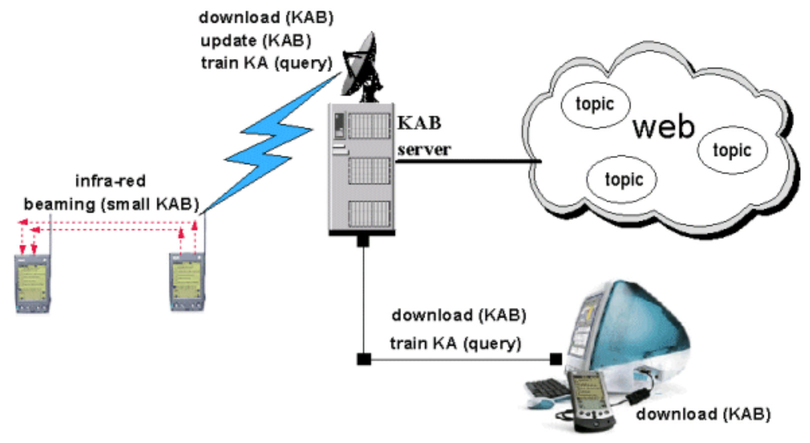


Figure 1: Server/Device Interaction.

small set of (usually 4-6) queries which pertain to the topic in question. Essentially, most of the pages in the set should be relevant to one or more of the queries.

Having answered the first two questions, we move on to tackle the type of knowledge which should be acquired by the agent. We claim that the agent can limit itself to acquiring the following information:

- A topic-specific lexicon, containing the dominant terms and lexical affinities of the domain (where lexical affinities are terms sharing a lexical relation and are identified as pairs of terms frequently co-occurring in close proximity [11]). As will be explained in Section 3, this lexicon is used both for query formulation assistance and as the source for corpus statistics that are required for the disconnected search process.
- In-depth knowledge (the full text) of the top-notch Web resources on the topic. These resources, which will be termed *the core set*, include both authoritative pages on the topic (authorities) and hubs (pages which are themselves resource lists on the topic). See [9] for more precise definitions of hubs and authorities. Many typical queries on the topic will have their answers found in the core set of pages, and by indexing their full text, the agent will be able to satisfy these queries in disconnected mode.
- Some knowledge of the second-tier of Web resources on the topic. Obviously, in order to keep the footprint of the KAB small, the size of the core set is quite limited and many valuable pages related to the topic will have been left outside of this set. However, while we may not be able to store the entire contents of these second-tier resources, it would be helpful to have some idea regarding their content, and to have the ability to suggest to users (in response to certain queries) that these URLs might be worth browsing at the next available moment of Internet connection.

Before moving on to detail the knowledge acquisition process, the key observation to be made is that the above knowledge is all implied by the core set of pages (and their associated text). Since the core pages are of the best available resources on the topic, their text should be representative of the domain language. It should contain topic-specific terminology, lexical affinities, and phrases. As for the required knowledge of second-tier pages, exactly such knowledge is contained in the hyperlinks which emanate from core pages. These links are associated with anchor text, which is often highly descriptive of the destination URL's contents. This is especially true for high quality pages, where the anchor text, like most other text, is highly informative. Moreover, since we aim to have top-notch hubs in the core set, their out-going links may be quite numerous and point to pages of high quality.

We call the URLs which are referenced by the core set *satellite pages*, and our experience shows that for core sets which contain about 100 pages, some 3000 satellite pages are referenced, with descriptive anchor text existing for many of these pages. Thus, the agent attains some knowledge on a much wider set of pages than it is actually able to store.

2.2 Knowledge Acquisition

Knowledge is acquired by collecting the core pages in an iterative process, governed by queries. Recall that the agent's creator defines the topic by a series of queries, which we will now denote by $q_1, \ldots, q_n$. Denoting the core set following the i'th query by C_i and C_s, we have $C_{i+1} = f_{ka}(C_i, q_i)$ where f_{ka} is the knowledge agent's core-set evolution-by-queries function. This function performs the process described below. Note that the initial core set, C_0, can be empty or contain some creator-specified URLs.

The core set evolves by processing the series of training queries, two types of which are supported by the system:

1. Text queries, which are keyword-based such as queries typically submitted to general purpose Web search engines. The agent will evolve its core set by searching for highly relevant resources to the query.
2. Sample URL queries, which specify a few (typically 1-5) seed urls, and whose purpose is to find quality pages which are closely related to the seeds.

Thus, for each training query, the agent performs a Web search. For text-based queries, some general purpose Web search engines (of the agent's creator choice) are presented with the query, returning a *root* set of candidate pages. For sample URL queries, the user supplied URLs are the root candidates. The collection of root candidates is expanded into a larger set of candidate pages, S, by following the hyperlinks surrounding the root pages. The pages which currently reside in the core set are also added to S. The resulting set of candidate pages, represents a directed subgraph of the WWW, whose nodes are the candidate pages and whose edges are the hyperlinks which connect the candidates.

The large set of candidate pages S are ranked by a combination of link-structure analysis and textual analysis. These analyses complement each other towards finding the pages which are both central to the agent's domain and satisfy the query well. The various components of the score are briefly described below.

1. The text analysis assigns each candidate page the following two separate cosine measure similarity scores, which attempt to grasp the extent to which the candidate answers the query, along with its general relevancy to the topic at hand.
   - The similarity between the page's text and the text of the query. This is straightforward for text queries, and is adapted for sample-URL queries by regarding the aggregate text of the sample URLs as the text of a pseudo query.
   - The similarity between the page's text and the domain's terminology. The domain's terminology is taken as the aggregated text of the pages which currently occupy the core set.
2. Analysis of the link structure of the subgraph S assigns each candidate page with a hub score and an authority score. These scores are computed using a combination of Kleinberg's mutual reinforcement algorithm [9] and SALSA [10], a stochastic link analysis algorithm. Following [5], links are weighted according to both the anchor text which is associated with them (and its similarity to the query), and according to their endpoints' membership in the evolving core set.

The textual and link analysis scores are combined to yield the overall *query relevance* score for each candidate page. At this stage, the core set can be updated. One of the innovations of the KA model is that the query relevance score does not exclusively govern whether a page is stored in the KAB. Rather, with each current member of the core set, there is an associated *fitness* score, which reflects that page's relevance to the domain through the course of the previous iterations of the agent's training phase. The fitness scores of the current core set members are compared against the query relevance scores of the new candidate pages. Pages compete for the right to be included in the KAB using an *evolutionary adaptation* mechanism. Briefly, the first few iterations see the core set grow until it reaches its maximal size, in terms of number of pages, as defined by the agent's creator. Once the core set is full, subsequent iterations cause pages with low domain fitness scores to become stale and be removed from the KAB

At the end of the training process, the KAB comprises a set of Web pages (and their associated fitness scores) that can be thought of as a set of category pages in some directory service. Unlike the latter though, the KA topic can be of any granularity and reflects the personal interests of its creator. These interests are not necessarily covered, or at least not in sufficient depth or specificity, by directory services. In addition, the KAB pages induce, as mentioned previously, a domain specific vocabulary, and a set of satellite pages.

While the training itself may take a few hours (depending mainly on the network connection available) since several hundred Web pages are crawled as part of this stage, it can be set up in a few minutes using a simple interface.

3. INFORMATION DISCOVERY IN DISCONNECTED MODE

As mentioned above, once the KAB is resident on the device, there is no need to access the server for many further information discovery tasks as they are conducted locally on the device.

The first discovery task that is conducted on the device is topic browsing and exploration. The KAB pages which are ranked by their fitness to the topic provide valuable information in the form of specialized bookmarks. Simply knowing the key Web pages on a given topic is a great source of information as demonstrated by directory services. They are available for simple browsing as classical bookmarks but can also be conveniently searched as explained below. In addition to reading the top resources, the top terms in the KAB can also be explored. These terms are most likely good starting points for searches in the domain of the KAB.

The second discovery task is topic-specific information retrieval which involves the following three stages:

1. **Query formulation:** Since we concentrate here on PDAs operated typically by stylus, or soft keyboard, word completion is not as crucial as on WAP phones for instance, yet it is a convenient feature as long as it is not too intrusive. Constantly changing the input string as characters are entered, as done in T9 [19], might be disorienting when using graffiti as users might think their graffiti was not properly recognized. Displaying optional completions, that change as input is entered seems like a reasonable solution that leaves quality control in the hand of the user. We suggest applying this approach not only at the word level but at the entire query level, by suggesting optional terms for completing and disambiguating queries as a word is completed (the end of a word being indicated by a space at the end of a string). Our key contribution in supporting query formulation assistance is 1) to use a topic-specific lexicon for both word and query completion as provided by the KAB and 2) to store the lexicon on the device for almost instantaneous response time. Once the query is formulated possibly after several refinement iterations, the search stage occurs.
2. **Search:** The entire search phase is conducted exclusively on the KAB. The KAB pages are indexed on the server during the training phase using a full-text search engine. Recall that the KAB consists of core pages and satellite pages as described in Section 2. While the entire text of each core page is inserted into the index, only the anchor text which described each satellite page in the referring core page is inserted into

the index. The index is downloaded to the device as part of the KAB. Given a query, the inverted index is used to locate the KAB pagess that best answer this query. The results are ranked based on the similarity to the query as well as their fitness score. Since more qualitative pages in the KAB have higher fitness scores, the ranking of the results will reflect the similarity to the particular query as well as the page's authority in the KABs topic. Due to the locality of the inverted index, ranked search results are returned to the user in less than 1s response time on a typical Palm V device.

3. **Browse results:** Result browsing is separated into high-level and low-level results. The high-level results include only the title of the page in order to fit as many results as possible on the small device's screen. Note that in the case of satellite pages, the title is actually the anchor text of the referring page as the page itself is unavailable on the device. Next, more detailed information can be displayed for each result. This includes query-focused snippets for the core KAB pages, and the actual URLs for the satellite pages. Finally, the text of resident pages can be displayed in disconnected mode, while the satellite pages (as well as the original core pages) can be browsed only if using a wireless PDA and consequently may require a few seconds of waiting until they are retrieved from the Web. We believe however that this delay is acceptable because it is performed typically at the end of the search process and is not reiterated too often.

In the remainder of this section we provide more detail regarding topic-specific query formulation and the disconnected search process.

3.1 Query Formulation Assistance

Word completion

In addition to simplifying text input, word completion can be helpful in spelling terms correctly. In the spirit of source code editors that provide word completion based on the limited set of terms in the the programming language vocabulary, we provide word completion based on the domain-specific vocabulary encapsulated in the KAB. As keys are pressed, the agent suggests the most frequent words in the KAB vocabulary consistent with the input prefix. This list is sorted in decreasing order of frequency, increasing the chances that the first term suggested is what the user had in mind. The user can thus complete query terms with one click rather than many keystrokes. See Section 4 for examples of word completions using some sample KABs.

Query completion

Automatic query completion is usually performed by suggesting terms related to the query terms using a global semantic word network such as WordNet (e.g., [20]). In our case, on the other hand, query completion is based on the KAB's local vocabulary which characterizes the domain's ontology and thus relations between terms are domain dependent. As a result, added terms disambiguate the query terms in the context of the specific domain.

The query completion process works by first clustering the terms in the query so that each cluster contains terms that most likely constitute a phrase and then identifying the best candidate expansion terms for each such phrase separately. The list of terms presented to the user is the union of each cluster's expansion terms. Since there is an upper bound on the total number of terms that can be suggested to the user (due to the small display size), each cluster is allotted the number of terms that it can contribute to this list based on the relative weight (in terms of frequency in the KAB) of the terms in the cluster. Term clustering as well as term suggestion is based on the lexical affinity relation. Recall that term t_2 is considered a lexical affinity of t_1 if t_2 is found in proximity (e.g., within 5 words) to t_1. The details of the query completion algorithm are described in Figure 2.

```
**Complete**$(q, L(KAB), n)$
  *input:*
    $q$ - the user's query
    $L(KAB) = \{(t, w(t), LA(t))\}$ where:
      $t$ is a term
      $w(t)$ its normalized frequency in the KAB
      $LA(t) = \{(t_i, w(t, t_i))\}$
        $(t, t_i)$ is a lexical affinity
        $w(t, t_i)$ its normalized frequency in the KAB
    $n$ - the maximum number of terms for completion

  for each $t \in q$, retrieve $(t, w(t), LA(t))$ from $L(KAB)$
  create a graph $G = (V, E)$ where:
    $V = \{t \in q \text{ associated with } w(t)\}$
    $E = \{(t_i, t_j) \text{ associated with } w(t_i, t_j) | t_i, t_j \in V\}$
      $(t_j, w(t_i, t_j)) \in LA(t_i)$
  compute the connected components of $G$, $C_1, \ldots, C_n$
  eList $\leftarrow \emptyset$
  $W \leftarrow 0$
  for each component $C_i = \{t_1, \ldots, t_k\}$
    $w(C_i) \leftarrow 1/k \sum_{j=1}^{k} w(t_j)$
    $W \leftarrow W + w(C_i)$
    Compute $LA(C_i) = \{(t, w'(t))\}$ where:
      $t \in \bigcap_{i=1}^{k} LA(t_i)$
      $w'(t) = 1/k \sum_{j=1}^{k} w(t, t_j)$
    sort $LA(C_i)$ in decreasing order according to $w'(t)$
    expand eList with the first $\frac{w(C_i)}{W} * n$ terms from $LA(C_i)$
  return eList
```

Figure 2: Query completion algorithm.

For example, consider the query "Circus dancer in Paris museums" and a KAB specializing in the French painter Toulouse-Lautrec. The terms "circus" and "dancer" are clustered together and the suggested terms in decreasing order of weights are "actress, performer, prostitute, moulin, rough, valentine, goulue". The terms suggested for "Paris" are "dorsay, musee, france, montmartre", and the term suggested for "museums" is "art". Assuming the number of suggested terms is limited to 9, and based on the computed cluster weights, the final list is "actress, performer, prostitute, moulin, rouge, dorsay, musee, art".

3.2 Disconnected Search

As mentioned above, the KAB pages are indexed during the training phase using a text search engine. We use Palm Pirate (Palm Information Retrieval Application for Text sEarch) [15], a search engine for the Palm developed at the IBM research lab in Haifa. The system allows users to store, search and view text collections on their Palm device. Palm Pirate is composed of an indexing component which runs on the desktop and a search component which runs on the Palm. The KAB pages are indexed using this indexing component during the training phase. The index is then downloaded to the device as part of the KAB.

The search component of the Palm Pirate uses a standard tf-idf [17] based scoring mechanism. The word statistics stored as part of the KAB's lexicon serve as the source for term frequencies used by this algorithm. The score for a KAB page is a combination of its textual similarity score to the query and its fitness score. Recall that the fitness score of a KAB page reflects its relevance to the training queries and therefore to the KAB's domain. By using the fitness score, authoritative domain pages within the collection of search results are given priority in terms of ranking.

4. SAMPLE SCENARIOS AND RESULTS

In this section, we present some scenarios that demonstrate several pervasive information discovery tasks being performed using our system. As mentioned in the introduction the premise is that that all the user has during the search and explore stage is a PDA and a modem or some other form of wireless communication. For example, he could be in the airport or on an airplane, in a museum, waiting for his child at soccer practice, or shopping.

We have created KABs on various topics ranging from technical fields such as Java, XML, Error Correcting Codes (ECC), to more recreational fields such as Toulouse-Lautrec, Broadway, Main Course Recipes (MCR), and Pokemon. We present one running scenario using the Toulouse-Lautrec KAB, as well as additional examples and statistics using some of the other KABs.

4.1 Agent Creation

Agents can be created from the desktop or the PDA. We offer both a "one-click" and an advanced interface. In addition, ready-made agents can be downloaded from an agent repository, or beamed from another PDA.

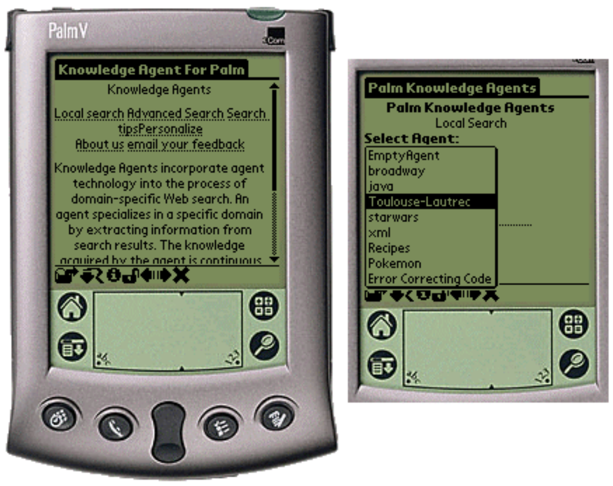
In the one-click interface, the user defines the agent by providing an optional list of seed URLs and 4-6 queries, which define the domain. In addition, the user must provide a name and description for the agent as well as their email address, which is used to identify them as the owners of the agent. In the advanced interface, users can create the agent in a more iterative fashion. They can inspect the results after each query, add new queries to agents that have already been trained, and add/remove pages from the KAB.

In Table 1 we present for four of our topics, the queries that we used to train an agent on the topic, the size of the corresponding KAB, the number of KAB pages, the precision of these pages (percent of pages relevant to the topic), the number of pages in the most closely related directory in Yahoo! [22], and the Open Directory as provided by Google [7].

All of the agent training was done based solely on sample queries. We did not provide the agent with any seed pages since we wanted to assume the least involvement in terms of the user in the knowledge acquisition stage. Our experience has shown that the hub and authority based techniques used in training enable the agent to acquire the most prominent pages pertaining to the topic. In terms of precision, note that both the Java and Recipes agents have extremely high precision. Only five of the 100 pages, in the Java KAB are not pure Java pages (they are more general programming language pages), and only one out of 100 pages in the MCR KAB is not related to main course recipes. The precision of the ECC and Toulouse - Lautrec agents albeit being slightly lower, is still relatively high ($\sim 85\%$). Note that the non-relevant pages could be removed using our KAB editing functionality. However, in the spirit of assuming as little user involvement as possible, we left them in the KAB to demonstrate that even if some of the pages in the KAB are not relevant to the topic, this does not hurt the functionality of the system. In terms of the precision of the satellite pages in the KAB, an empirical study based on sampling showed that on average 75% of the satellite pages are relevant to the domain of the KAB.

In comparison to commercial Web directories, only Java and Toulouse - Lautrec have corresponding directory entries in Yahoo! and in the Open Directory project. The Java category comprises many pages in both directories (343 and 1478 pages, respectively) and consequently would be too large to cache on a small device. The Toulouse - Lautrec directory, on the other hand, has very few pages and thus does not contain enough information to be useful as a Toulouse - Lautrec representative repository. There are no direct corresponding directories for either the ECC or MCR KABs.

4.2 Topic-Specific Browsing and Exploration


PalmV
Knowledge Agent For Palm
Knowledge Agents
Local search Advanced Search Search tipsPersonalize
About us email your feedback

Knowledge Agents incorporate agent technology into the process of domain-specific Web search. An agent specializes in a specific domain by extracting information from search results. The knowledge

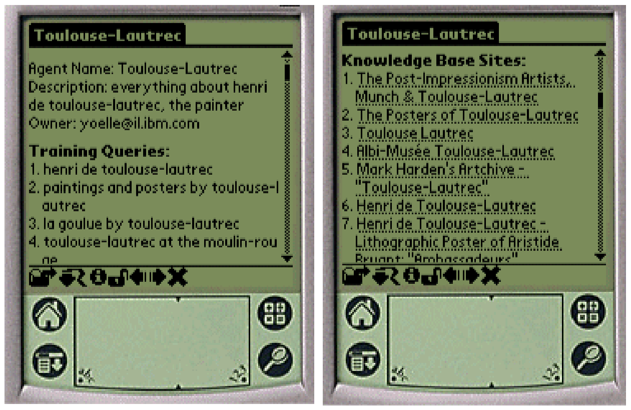
Figure 3: Selecting a KAB

In Figure 3 we see the Toulouse - Lautrec KAB being selected from a pull-down list that contains the names of all the KABs available on the device.

Once an agent is selected several operations are available which allow the user to explore the content of the KAB. Figure 4 depicts this functionality. The left screenshot shows the user simply viewing some basic information regarding the agent such as the name and description that the owner gave it as well as the queries used for training it. This information provides some feeling for the scope of the agent, and may be of interest when acquiring a new agent via downloading or beaming. The right screen shot shows the KAB pages sorted by their fitness score. That is, the first pages in the list are the top hubs or authorities in the domain. The screenshot in Figure 5 shows the top terms of the KAB. These terms are most likely good starting points for searching the KAB, and indeed clicking on any of them will initiate

Agent	Training Queries	KAB size	# of KAB pages	Open Directory	Yahoo
Java	java programming language java class libraries java jdk java compiler java beans	1.86MB	100 precision: 95%	1478	343
Error Correcting Code	error correcting codes linear codes convolution codes turbo codes BCH codes	1.25MB	100 precision: 84%	closest category coding theory (28)	no closest category
Toulouse - Lautrec	henri de toulouse-lautrec paintings and posters by toulouse-lautrec la goulue by toulouse-lautrec toulouse-lautrec at the moulin-rouge	430kb	60 precision: 87%	1	7
Main Course Recipes	chicken recipes beef recipes fish and seafood recipes pork recipes lamb recipes	500KB	99 precision: 99%	closest category recipe collections (486)	closest category (among 8) recipes (1057)

Table 1: KAB statistics.


Toulouse-Lautrec

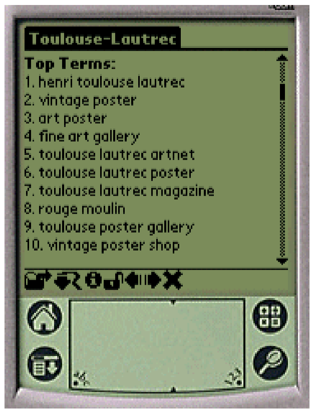
Agent Name: Toulouse-Lautrec
Description: everything about henri de toulouse-lautrec, the painter
Owner: yoelle@il.ibm.com

Training Queries:
1. henri de toulouse-lautrec
2. paintings and posters by toulouse-lautrec
3. la goulue by toulouse-lautrec
4. toulouse-lautrec at the moulin-rouge

Toulouse-Lautrec

Knowledge Base Sites:
1. The Post-Impressionism Artists, Munch & Toulouse-Lautrec
2. The Posters of Toulouse-Lautrec
3. Toulouse Lautrec
4. Albi-Musée Toulouse-Lautrec
5. Mark Harden's Artchive - "Toulouse-Lautrec"
6. Henri de Toulouse-Lautrec
7. Henri de Toulouse-Lautrec - Lithographic Poster of Aristide Bruant: "Ambassadeurs"

Figure 4: KAB exploration


Toulouse-Lautrec

Top Terms:
1. henri toulouse lautrec
2. vintage poster
3. art poster
4. fine art gallery
5. toulouse lautrec artnet
6. toulouse lautrec poster
7. toulouse lautrec magazine
8. rouge moulin
9. toulouse poster gallery
10. vintage poster shop

Figure 5: Top terms in KAB

a search for pages that contain these terms.

4.3 Query formulation

Figure 6 demonstrates the word completion functionality. After typing just two letters "am" and hitting the refine button only two words are suggested: ambassadeur, and american. For "amb" there is only one completion - ambassadeur. In contrast, the Broadway KAB would suggest "american, amadeus, amateur, amazing" for the letters "am", while there are no words suggested for "amb".

As another example, consider the two letters "pi": The Pokemon KAB suggests the words: pikachu, picture, pinball, pikablu; The ECC KAB suggests: pietrobon, pipeline, pinch, pixel; The MCR KAB suggests: pie, piece, pineapple, pizza, pink. Clearly, no general purpose word completion could achieve such results.

Figure 7 shows the phrase completion functionality. After selecting the word ambassadeur from the word completion list and hitting refine again, the system suggests additional terms for the query. In this case the terms are: bruant, aristide, poster, lautrec, toulouse. The user selects bruant and aristide, to complete the query formulation stage.

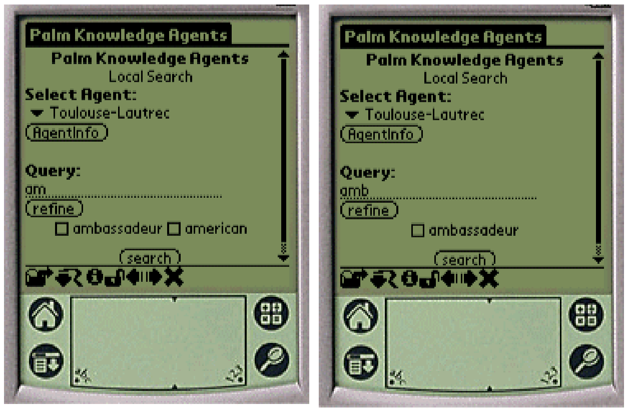
Some additional examples of query completion are provided in Table 2.

4.4 Disconnected Search

Figure 8 presents the results for the query "ambassadeur aristide bruant". The left image shows the first result screen. In this screen, only the titles of the top results are displayed (with no urls or snippets) in order to fit as many results as possible on the small screen. Once the user clicks on one of these titles, more information about this particular result is presented. More specifically, if the result is one of the core pages (for which we cache the text of the page locally), the title as well as a query focused snippet of the result with the

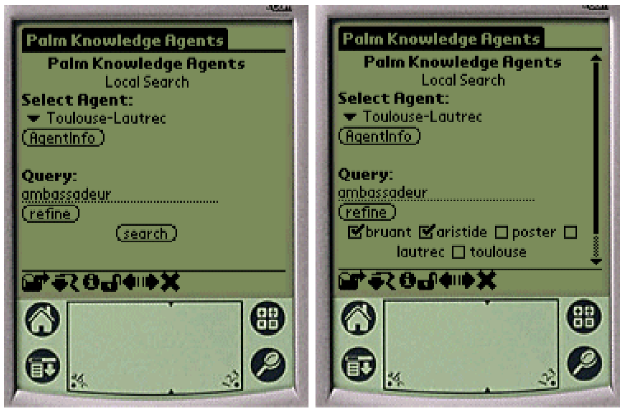
Agent	Query term	Query completion
Java	bean	java, pure, awesome
Java	java bean	program, faq, tutorial
Recipes	bean	white, lamb, free, middot, rice
Pokemon	card	trade, game, pokemon
Java	card	game, java, applet
Java	free	java, software, tool, download
ECC	free	code, distance, hamming
Toulouse - Lautrec	free	poster, download
Pokemon	free	pokemon, card, email, game
Recipes	red	wine, dark, slightly, vinegar
Pokemon	red	blue, pokemon, yellow, green,

Table 2: Query Completion Examples


Palm Knowledge Agents
Palm Knowledge Agents
Local Search
Select Agent:
▼ Toulouse-Lautrec
AgentInfo
Query:
am
refine
☐ ambassadeur ☐ american
search

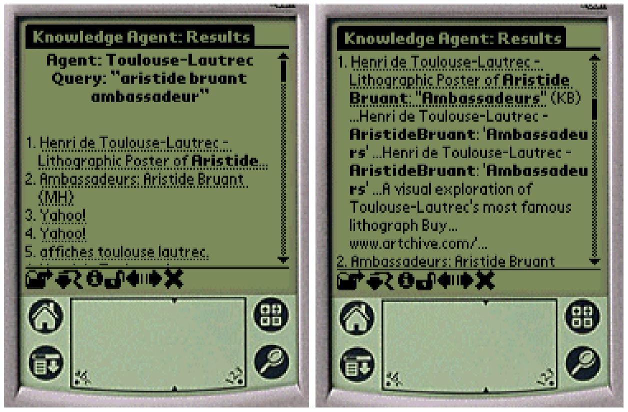
Palm Knowledge Agents
Palm Knowledge Agents
Local Search
Select Agent:
▼ Toulouse-Lautrec
AgentInfo
Query:
amb
refine
☐ ambassadeur
search

Figure 6: Topic-specific word completion


Palm Knowledge Agents
Palm Knowledge Agents
Local Search
Select Agent:
▼ Toulouse-Lautrec
AgentInfo
Query:
ambassadeur
refine
search

Palm Knowledge Agents
Palm Knowledge Agents
Local Search
Select Agent:
▼ Toulouse-Lautrec
AgentInfo
Query:
ambassadeur
refine
☑ bruant ☑ aristide ☐ poster ☐ lautrec ☐ toulouse

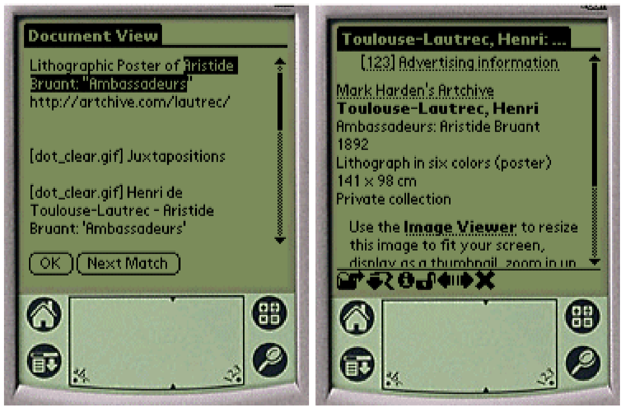
Figure 7: Topic-specific phrase completion


Knowledge Agent: Results
Agent: Toulouse-Lautrec
Query: "aristide bruant ambassadeur"

1. Henri de Toulouse-Lautrec - Lithographic Poster of **Aristide**...
2. Ambassadeurs: Aristide Bruant (MH)
3. Yahoo!
4. Yahoo!
5. affiches toulouse lautrec.

Knowledge Agent: Results
1. Henri de Toulouse-Lautrec - Lithographic Poster of **Aristide Bruant**: "**Ambassadeurs**" (KB)
...Henri de Toulouse-Lautrec - **AristideBruant**: '**Ambassadeurs**' ...Henri de Toulouse-Lautrec - **AristideBruant**: '**Ambassadeurs**' ...A visual exploration of Toulouse-Lautrec's most famous lithograph Buy...
www.artchive.com/...
2. Ambassadeurs: Aristide Bruant

Figure 8: Search results for "Aristide Bruant Ambassadeur"


Document View
Lithographic Poster of Aristide Bruant: "Ambassadeurs"
http://artchive.com/lautrec/

[dot_clear.gif] Juxtapositions

[dot_clear.gif] Henri de Toulouse-Lautrec - Aristide Bruant: 'Ambassadeurs'

OK Next Match

Toulouse-Lautrec, Henri: ...
[123] Advertising information

Mark Harden's Artchive
Toulouse-Lautrec, Henri
Ambassadeurs: Aristide Bruant
1892
Lithograph in six colors (poster)
141 x 98 cm
Private collection

Use the **Image Viewer** to resize this image to fit your screen, display as a thumbnail, zoom in up

Figure 9: Scanning selected results

query terms highlighted is displayed (right image). On the other hand, if the result corresponds to a satellite/referenced page, we show the anchor text of the reference as the title and the url of the page itself.

Moving along to Figure 9, the user has now chosen to access the result page by clicking on the title of the desired result. In case of core pages, the text of the page is retrieved from the cache. The text can be browsed with query terms highlighted (left image). Satellite pages need to be retrieved from the Web server where they reside by wireless communication. Note that this is the first time that network access is required. All of the steps described in the sample scenario so far were performed in disconnected mode.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a focused-search approach specifically designed for the mode of work and the constraints dictated by pervasive devices. We first identified the constraints inherent to pervasive devices as well as their implication on the search experience from such devices. We then argued that topic-specific focused search is a key to overcome these limitations. More specifically, we claimed that by coupling focused search with a knowledge representation that encapsulates domain knowledge in a concise yet representative fashion, it is possible to make information discovery from pervasive devices practical as well as efficient.

We suggested a representation for topic-specific information based on "knowledge-agent bases" (KABs) where topics are represented and stored in a persistent repository with a very small footprint which can be downloaded to small devices. While knowledge acquisition is done on a server, by downloading KABs to the device, many information discovery tasks can be performed in disconnected mode. The KAB encapsulates the information necessary to access information about a topic such as key concepts and key Web pages, to assist in query formulation and to perform topical searches on the device itself. Using this model, response time for many tasks is much better than in previous connected models.

With more mobile knowledge seekers turning to pervasive devices on a daily basis, the importance of improving the usability of information discovery tasks on such devices will become even more crucial in the future. While we have focused mainly on PDAs and in particular on PalmOS based devices in this paper, the future is clearly in mobile phones that have a much larger potential market. We believe that several aspects of our model are applicable and may be even more critical in this arena. Form factor, for example, imposes even heavier constraints and the query formulation assistance stage is even more crucial. Topic-specific knowledge encapsulation could thus result in even greater benefits. While current mobile phones do not have enough memory or processing power to use our KAB model as is, hopefully these devices will become more powerful and more importantly will open their OS so users can customize and exchange information.

Acknowledgments

We thank Miki Herscovici and Doron Cohen for providing the Palm Pirate application, and their expertise on PalmOS. We also thank Uri Weiss and Roni Raab for their contribution to the codebase and Rob Farrell for his feedback on user's experience.

Vitae

Yariv Aridor is a research staff member at the IBM Research Laboratory in Haifa, Israel. He received his MSc. and PhD. in Computer science from the Tel-Aviv university, Israel, in 1989 and 1995, respectively. His research interests include distributed systems, mobile object-oriented systems and agent technology.

David Carmel is a Research Staff Member at the IBM Research Laboratory in Haifa, Israel, and belongs to the *Information Retrieval and Organization* group. His research interests include information retrieval, multi-agent systems and artificial intelligence. He received his MsC and PhD in Computer science from the Technion, Israel institute of technology, in 1993 and 1997 respectively.

Ronny Lempel is a Ph.D. Student in the faculty of Computer Science, Technion, Haifa, Israel, focusing on WWW link-structure analysis. He received his B.Sc. and M.Sc. from the same faculty in 1997 and 1999, respectively.

Yoelle S. Maarek is a Research Staff Member at the IBM Research Laboratory in Haifa, Israel and manages the *Information Retrieval and Organization* group. Her research interests include information retrieval, Internet applications, and software reuse. She graduated from the *Ecole Nationale des Ponts et Chaussees*, Paris, France, as well as received her D.E.A (graduate degree) in Computer Science from Paris VI University in 1985. She received a Doctor of Science degree from the Technion, Haifa, Israel, in January 1989.

Aya Soffer is a Research Staff Member in the *Information Retrieval and Organization* group at the IBM Research Laboratory in Haifa, Israel. Her research interests include pictorial information systems, information retrieval, and nontraditional database systems. She received her MsC and PhD degrees in Computer science from the University of Maryland at College Park in 1992 and 1995, respectively.

6. REFERENCES

[1] Y. Aridor, D. Carmel, R. Lempel, Y. Maarek, and A. Soffer. Knowledge agents on the web. In *Proceedings of the 4th International Workshop on Cooperative Information Agents, CIA 2000, LNAI 1860*, pages 15–26, Boston, MA, July 2000. Springer.

[2] AvantGo. http://www.avantgo.com.

[3] I. Ben-Shaul, M. Herscovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, V. Soroka, and S. Ur. Adding support for dynamic and focused search with fetuccino. In *Proceedings of the 8th International Word Wide Web Conference (WWW8)*, pages 575–587, Toronto, Canada, May 1999. Elsevier.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Focused web searching with PDAs. *WWW9 / Computer Networks*, 33 (1-6):213–230, 2000.

[5] S. Chakrabarti, B. Dom, D. Gibson, J. M. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource list compilation by analyzing hyperlink structure and associated text. *WWW7 / Computer Networks*, 30(1-7):65–74, 1998.

[6] S. Chakrabarti, B. Dom, and M. van den Berg. Focused crawling: A new approach to topic-specific web resource discovery. *WWW8 / Computer Networks*, 31 (11-16):1623–1640, 1999.

[7] Google. Inc. http://www.google.com/.

[8] E. Kaasinen, M. Aaltonen, J. Kolari, S. Melakoski, and T. Laakko. Two approaches to bringing internet services to WAP devices. *WWW9 / Computer Networks*, 33 (1-6):231–246, 2000.

[9] J. M. Kleinberg. Authoritaive sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 25-27, pages 668–677, January 1998.

[10] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *WWW9 / Computer Networks*, 33 (1-6):387–401, June 2000.

[11] Y. Maarek and F. Smadja. Full text indexing based on lexical relations, an application: Software libraries. In N. Belkin and C. van Rijsbergen, editors, *Proceedings of SIGIR89*, pages 198–206. Cambridge, MA, ACM press, 1989.

[12] T. Masui. An efficient text input method for pen-based computers. In *Proceedings of the CHI 98 Conference on Human Factors in Computing Systems*, pages 328–335, 1998.

[13] S. Mukherjea. Wtms: A system for collecting and analyzing topic-specific web information. *WWW9 / Computer Networks*, 33 (1-6):457–471, 2000.

[14] Palm. http://www.palm.net.

[15] Pirate Search for Palm. IBM research lab in Haifa. `http://www.alphaworks.ibm.com/tech/piratesearch`.

[16] Research in motion. http://www.rim.net.

[17] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. Computer Series, McGraw-Hill, New York, 1983.

[18] J. R. Smith, R. Mohan, and C. S. Li. Transcoding internet content for heterogeneous client devices. In *Proceedings of the IEEE International Symposium on Circuits Systems (ISCAS), Special session on Next Generation Internet*, June 1998.

[19] Tegic t9 text input. http://www.t9.com.

[20] E. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-93)*, pages 171–180, Pittsburgh, 1993.

[21] Wap forum. http://wapforum.org.

[22] Yahoo Inc. Yahoo! http://www.yahoo.com.