Navigating the Intranet with High Precision

Huaiyu Zhu
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
huaiyu@us.ibm.com

Alexander Löser*
SAP Research CEC Dresden
Chemnitzer Str. 48, D-01187 Dresden, Germany
alexander.loeser@sap.com

Sriram Raghavan
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
rsriram@us.ibm.com

Shivakumar Vaithyanathan
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
shiv@almanden.ibm.com

ABSTRACT

Despite the success of web search engines, search over large enterprise intranets still suffers from poor result quality. Earlier work [6] that compared intranets and the Internet from the view point of keyword search has pointed to several reasons why the search problem is quite different in these two domains. In this paper, we address the problem of providing high quality answers to navigational queries in the intranet (e.g., queries intended to find product or personal home pages, service pages, etc.). Our approach is based on offline identification of navigational pages, intelligent generation of term-variants to associate with each page, and the construction of separate indices exclusively devoted to answering navigational queries. Using a testbed of 5.5M pages from the IBM intranet, we present evaluation results that demonstrate that for navigational queries, our approach of using custom indices produces results of significantly higher precision than those produced by a general purpose search algorithm.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms

Algorithms, Measurement, Experimentation.

Keywords

Intranet search, enterprise search, High precision information retrieval

1. INTRODUCTION

The ultimate goal of any search system is to answer the need behind the query. Analogous to Broder's taxonomy of Web search, queries on the intranet can also be classified as informational, navigational or transactional [1]. In this paper, we present our approach for spotting and resolving "*navigational queries*" in the intranet. The purpose of such queries is to reach a particular page that the user has in mind, either because they visited it in the past or because they assume that such a page exists. We refer to such pages as "*navigational pages*". Typical examples of navigational pages in the IBM intranet include the homepage of a person, an authoritative page for a service such as Employee Assistance Program, the homepage of a department such as The Center for Business Optimization, or the internal home page for the DB2 UDB product.

Popular web-search engines routinely answer such navigational queries. For instance, if the user query is the name of a person, the top-ranked results from most search engine are predominantly user homepages. Unfortunately, this does not imply that navigational search in the intranet is a solved problem. Indeed, the differences between intranets and the Internet (from the view point of search) were aptly captured in four axioms presented in [6]. We draw particular attention to axioms 2 and 4 that we repeat below for convenience:

Axiom 2 A large fraction of queries tend to have a small set of correct answers (often unique) and the unique answer pages do not usually have any special characteristics.

Axiom 4 Large portions of intranets are not search-engine-friendly.

The first part of Axiom 2 essentially describes navigational queries and highlights the the difficulty in identifying such pages in the intranets. Axiom 4 refers to multiple challenges. For one, it refers to queries for company-specific information such as employees, projects, product names, acronyms, etc. It also reflects the difficulties encountered in a geographically disperse corporation with very local, organization-specific and site-specific terminologies and queries.

Consider, for example, a user seeking to find the "`Global Technology Services`" homepage. The most common query for this task is "`gts`"[1]. Using standard web-IR techniques — essentially a combination of link-analysis and IR ranking heuristics — the top 50 hits on the IBM intranet do not contain the correct answer[7]. On the other hand, *a priori*, if the Global Technology Services homepage had been identified and associated with the term

[*] Work performed while at IBM Almaden Research Center.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

[1] This is one of the top 4000 unique queries on the IBM Intranet.

"gts", the task of the search engine at query time is extremely straightforward. This is precisely what we set out to accomplish.

Specifically, our proposed approach consists of an offline process in which we recognize all navigational pages and associate appropriate term variants with each page. It turns out that during the actual process of identifying a navigational page we get more information than simply whether a page is navigational. In particular, depending on the sequence of analysis steps that is used to identify them, navigational pages are placed into one of several "*semantic buckets*" (e.g., there is a semantic bucket that holds all of the personal home pages). For each bucket, we build a standard inverted index using the terms and variants associated with the set of pages in that bucket — we refer to this index as a "*navigational index*". At runtime, a given search query is executed on all these navigational indices and the results are merged to produce the final answer to the navigational query.

Focus on precision. Since we are concerned only with navigational queries where the goal is to identify the one or few right answers to each query, precision is of the utmost importance. Our overall approach, a number of design choices, as well as the evaluation results that we present in the subsequent sections are guided by this emphasis on precision.

Contributions. The primary contributions of this paper are as follows:

- We propose a solution to the problem of answering navigational queries on the intranet. Our solution is based on off line identification of navigational pages, generation of term-variants to associate with each page, and the construction of separate indices exclusively devoted to answering navigational queries.
- We propose a generic templatized procedure for identifying navigational pages using a sequence of "*local*" (intra-page) and "*global*" (cross-page) analysis. We illustrate this template with several specific local and global analysis algorithms.
- We consider the problem of filtering and ranking the results of navigational queries based on user profiles. In this context, we present a technique for answering geo-sensitive navigational queries, i.e., queries for which the correct result page depends on the geography of the user posing the query.
- Using a collection of 5.5 million pages from the IBM intranet and a set of real intranet user queries issued within IBM during the period June–August 2006 as our test bed, we report evaluation results that validate the efficacy of our approach.

2. IDENTIFYING NAVIGATIONAL PAGES

As mentioned in Section 1, the first step in answering navigational queries is one of *identifying navigational pages*. Our strategy for identifying such pages broadly consists of two phases of analysis:

- *Local (page-level) analysis.* In the first phase, each page is individually analyzed to extract clues that help decide whether that page is a "*candidate navigational page*".
- *Global (cross-page) analysis.* In the second phase, groups of candidate navigational pages are examined to weed out false positives and generate the final set of navigational pages.

In this section, we present a generic template for local and global analysis. We instantiate this template by describing several specific local and global analysis algorithms. Our algorithms are motivated by observations about the naming and organization of intranet web pages. These observations, derived from extensive manual examination of the IBM intranet, appear to hold true across a range of web sites — from individual home pages to project, group, or departmental portals.

- *It is sufficient to restrict attention to specific attributes of a page* In general we observed that a small but specific set of attributes are sufficient indicators of a navigational page. We refer to such attributes as "*navigational features*". Examples of such features are title and URL. For instance, the presence of phrases such as "home", "intranet", or "home page" in the title, or an URL ending in "index.html" or "home.html", is a strong indication that the corresponding page is a candidate navigational page. (We note here that the importance of exploiting URLs and titles for intranet search was also recognized by the authors of [6].)
- *Page hierarchies provide input groups for global analysis.* While page-level cues yield candidate navigational pages, they also include a number of false positives. As an example, several pages in the IBM intranet related to "*Global Technology Services*" mention the phrase "global technology services intranet" in the title. However, not all of them are home pages. In fact, there is exactly one page — the "*root*" page of the global technology services web site — that is the preferred answer to any navigational query referring to global technology services. Later in this section, we describe how candidate pages can be grouped into hierarchies based on URLs or titles and fed into the global analysis phase to identify navigational pages.
- *Domain dictionaries can yield significant benefits.* Consistent with observation in [6] domain dictionaries such as acronyms and employee directories can dramatically improve precision. Acronyms, for example, proliferate throughout a modern enterprise as they are used to compactly name everything from job descriptions to company locations and business processes. Our experimental results (Section 6) show that the ability to match acronyms to their expansions significantly improved the performance of navigational search. A further proof point for the importance of acronyms is the fact 32 of the 60 most frequent search queries on the IBM intranet were acronyms.[2]

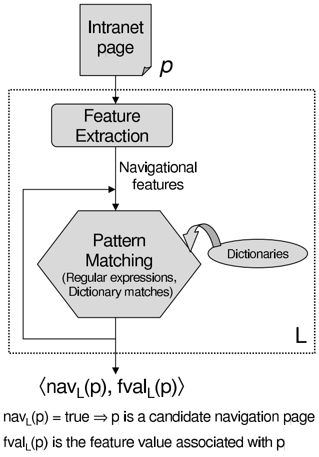

Intranet page
p
Feature Extraction
Navigational features
Pattern Matching
(Regular expressions, Dictionary matches)
Dictionaries
L
⟨nav_L(p), fval_L(p)⟩
nav_L(p) = true ⇒ p is a candidate navigation page
fval_L(p) is the feature value associated with p

Figure 1: Local Analysis Template

[2] Based on query logs gathered over a period of 3 months starting June 2006.

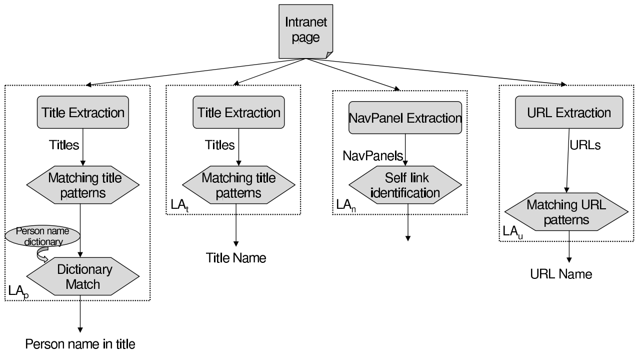


Figure 2: Local Analysis Algorithms

2.1 Local Analysis

Figure 1 shows the generic template of a local analysis algorithm. As shown in the figure, the first step of such an algorithm is feature extraction in which one or more navigational features are extracted from the input page. These navigational features are then fed into a sequence of pattern matching steps. Each pattern matching step either involves the use of regular expressions or an external dictionary (such as a dictionary of person names or product names). Depending on the output of the final pattern matching step, the local analysis algorithm will decide whether a given page is a "*candidate navigational page*" and optionally associate a "*feature value*" with each output candidate.

We use the following notations in this section: Given an intranet page p and a local analysis algorithm L, let $\mathrm{nav}_{\mathrm{L}}(p)$ be a boolean value that is set to true iff p is a candidate navigational page according to L. Let $L(\mathcal{P}) = \{p : \mathrm{nav}_{\mathrm{L}}(p)\}$ denote the set of candidate navigational pages identified by L from an input set of pages $\mathcal{P}$. For each $p \in \mathcal{P}$, let $\mathrm{fval}_{\mathrm{L}}(p)$ denote the "*feature value*" associated with p by L.

In the following, we describe four local analysis algorithms (Figure 2) that fit the template of Figure 1. Note that we use rounded rectangles to depict extraction steps and hexagons to depict pattern matching steps.

Candidate title home pages (LA_t). Algorithm LA_t is inspired by our earlier observation that page titles are excellent navigational features. LA_t extracts titles from web pages and applies a carefully crafted set of regular expression patterns to these titles. Examples of patterns that we used are given below (using the syntax of Java regular expressions [9]):

```
\A\W*(.*)’s? <Home>\b
\b<Home> of (.*)<junk>\Z
\A\W*(.*) <Home><junk>\Z
\A\W*(.*) Home<junk>\Z
\A\W*(.*) Info Page<junk>\Z
```

where

```
<Home>=(?:Home\s*Page|Intranet (?:Site|Page))
<junk>=[\W\d]*
```

Essentially, these patterns match titles that contain phrases such as "`John Smith’s home page`", "`Lenovo Intranet`", or "`Autonomic Computing Home`". When a match is detected, the corresponding page is returned as a candidate navigational page and a portion of the match is used as the feature value[3]. For our examples, the feature values are respectively "`John Smith`", "`Lenovo`", and "`Autonomic Computing`".

Candidate personal home pages (LA_p). Algorithm LA_p is an extension to LA_t that focuses only on personal home pages. For each page p such that $\mathrm{nav}_{\mathrm{LA_t}}(p) = true$, we apply an additional pattern matching step in which $\mathrm{fval}_{\mathrm{LA_t}}(p)$ is is matched against a dictionary of person names. A successful match results in p being returned as a candidate personal home page with the name of the person as the feature value. In our implementation, a dictionary of person names was produced by extracting names from *BluePages* — IBM's internal enterprise-wide employee directory.

Note that whenever other dictionaries of entity names are available (dictionary of product names, department names, names of company locations, etc.) a similar local analysis algorithm can be employed. The use of such dictionaries is a powerful mechanism for exploiting available organization-specific structured information. This helps to make local analysis algorithms more effective and customized to a particular intranet.

While the distinction between LA_p and LA_t (i.e., specialized processing of personal home pages) may seem artificial, it is important for reasons that will be clear when we describe term-variant generation in Section 4.1. Essentially, the process of generating term-variants for person names is quite unique and does not apply to all feature values in general. By keeping personal home pages separate, the appropriate variant generation technique can be applied exclusively to those pages.

Candidate URL home pages (LA_u). Analogous to the previous algorithms that operate on the title, algorithm LA_u applies regular expression patterns on the URL of a page to identify navigational cues. Our patterns captured several heuristics, two of which we list below as examples:

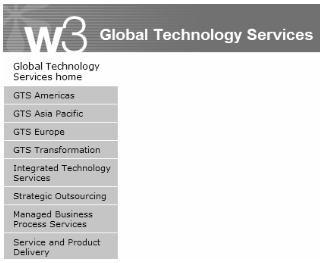
- When the URL ends in a "`/`" or contains "`index.html`", "`home.html`", "`logon.jsp`", etc., as the last segment, the corresponding page is returned as a candidate navigational page. The feature value is the last directory segment of the URL. For example, the URL `http://w3-03.ibm.com/services/iga/index.html` will result in the feature value "`iga`".
- When the URL only contains a host name or a host name followed by "`index.html`", "`index.jsp`", etc., the corresponding page is considered a candidate navigational page. The associated feature value is a segment of the host name after excluding common segments such as "`www`", "`w3`", "`ibm.com`", etc. For example, the URL `http://w3.research.ibm.com/` will result in the feature value "`research`".

In the interest of space, we omit the details of the fairly sophisticated regular expressions that we developed to reliably apply the above heuristics.

Candidate NavLink home pages (LA_n). Finally, algorithm LA_n exploits a particularly common characteristic of IBM intranet pages, namely, the presence of navigation panels[4]. Typically such panels are part of a department/group-level page template and contain a list of links that point to specific pages within the web site of the corresponding department or group (see Figure 3). The key

[3] The feature value is essentially the substring of the title that matches the portion of the regular expression within parentheses

[4] Such navigation aids of one form or another are quite common in most web sites, within and outside the enterprise.



w3 Global Technology Services

Global Technology Services home
GTS Americas
GTS Asia Pacific
GTS Europe
GTS Transformation
Integrated Technology Services
Strategic Outsourcing
Managed Business Process Services
Service and Product Delivery

Figure 3: Sample navigation panel

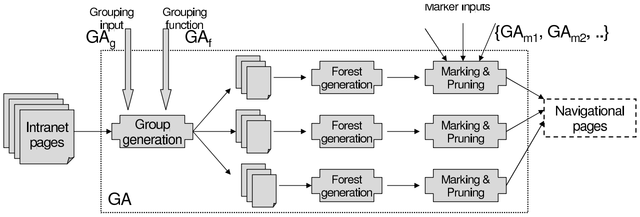


Figure 4: Site Root Analysis Algorithm

observations that allows us to exploit navigation panels is the following: whenever one of the links in a navigation panel on page p is associated with text such as "`home`" or "`main`" (e.g., the link labeled "`global technology services home`" in Figure 3) and if the link is a self-reference, (i.e., the link points to p), p is likely to be a navigational page.

2.2 Global analysis

The local analysis algorithms presented in the previous section rely on patterns in the title or URL of a page. Given multiple pages with similar URLs/titles that match these patterns, our local analysis algorithms will recognize all of these pages as candidate navigational pages and assign identical feature values. To illustrate, let us revisit our *Global Technology Services* example. Several pages within the Global Technology Services website contain the phrase "`Global Technology Services Intranet`" in the title. Therefore all of these pages will be identified by LA_t as candidate home pages with a feature value "`Global Technology Services`". Similarly, authors of personal home pages often use a title such as "`Howard Ho home page`" on all the pages in their website (CV page, publications page, projects page, etc.). As a result, LA_p identifies all of these pages as candidate personal home pages.

However, in both examples, the intended target of users issuing navigational keyword queries is the "*root*" or "*entry*" page. To filter out spurious pages from the output of local analysis, we propose a global analysis algorithm called "*site root analysis*".

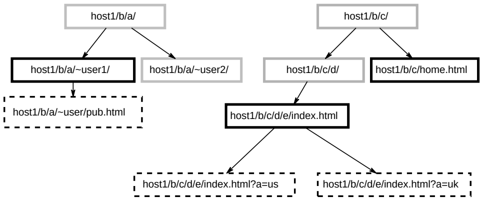


Figure 5: Forest generation

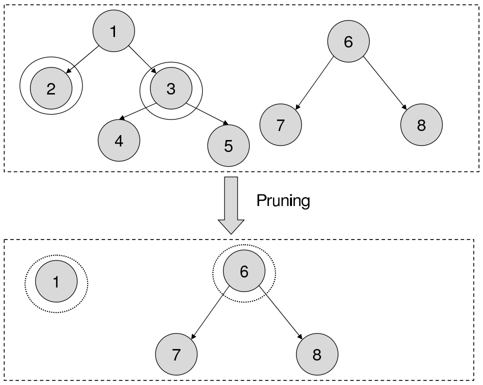


Figure 6: Marking and Pruning example

Site root analysis. Site root analysis exploits the hierarchical structure inherent in groups of related pages to identify root pages. Figure 4 shows a generic template for this algorithm. Specific instances of this algorithm can be employed by supplying a "*grouping input*", a "*grouping function*", and a set of one or more "*marker inputs*". In a typical instantiation, the grouping and marker inputs will be the set of candidate navigational pages identified by different local analysis algorithms. In the following, we first present the generic algorithm and then the describe specific instantiations shown in Figure 7.

For a particular instance GA, let GA_g, GA_f, and $\{GA_{m_1}, \ldots, GA_{m_n}\}$ denote the grouping input, grouping function, and marker inputs respectively. The first step in site root analysis, called "*group generation*", partitions the input pages in GA_g into groups of related pages. Precisely how this grouping is accomplished is dependent on GA_f and will be illustrated later in this section when we describe specific grouping functions.

For each group, the next step is to execute a process known as "*forest generation*". In this process, given a group S, we produce a graph G_S such that (i) there is a node in G_S for each page in S, and (ii) there is a directed edge from pages a to b iff the following condition holds: among all the URLs of pages in S, $\mathrm{url}(a)$ is the longest URL that is a strict prefix of $\mathrm{url}(b)$.[5] It is easy to see that the resulting graph G_S is a forest of rooted trees as illustrated by the example in Figure 5.

In the next step, every node in the forest whose corresponding page is present in at least one of the marker inputs $G_{m_1}, \ldots, G_{m_n}$ is "*marked*". Using these marked nodes, the final output of this algorithm is generated as follows:

- Fist, we add all marked nodes to the set of output navigational pages.
- Next, from each tree in the forest, we remove all subtrees that are rooted at marked nodes (i.e., all marked nodes and their descendants are removed).
- Finally, we add the root of each remaining tree in the forest to the output.

Figure 6 illustrates how marking and pruning works using a simple example of a forest with two trees. Let the nodes labeled 2 and 3 be marked as indicated by the circles around the node labels. The pruning step will remove the entire subtree rooted at 3 as well as

[5] When computing prefixes, we only consider complete URL segments. Thus "`http://host/abc`" is a prefix of "`http://host/abc/def`" but "`http://host/abc/de`" is not a prefix.

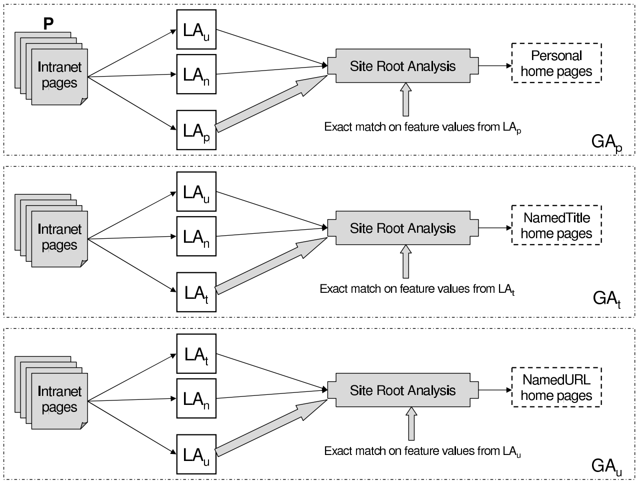


Figure 7: Specific instances of site root analysis

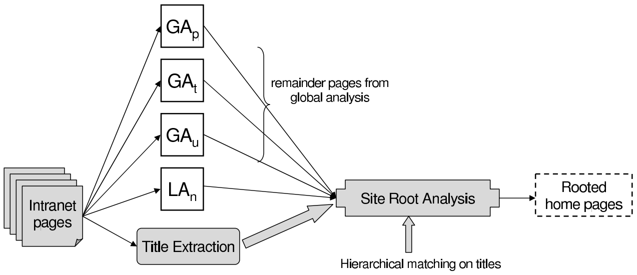


Figure 8: Site root analysis using hierarchical title grouping

the nodes 2 and 3 resulting in the forest shown in the bottom part of the figure. Nodes 1 and 6 are the roots of the remaining trees in the forest. Thus, the final output consists of nodes 2, 3, 1, and 6.

Instantiating site root analysis. Figure 7 shows three specific instantiations of site root analysis that we have implemented and deployed on the IBM intranet.

Algorithm GA_p is the global analysis used to identify Personal home pages. GA_p uses the output of LA_p as the grouping input and the output of LA_u (local analysis on URL) and LA_n (local analysis on navigation panel) as the marker inputs. Recall that LA_p is the local analysis that identifies candidate personal home pages and extracts person names from the title as feature values. Since GA_p uses an "*exact match*" grouping function, all candidate personal home pages with identical person names will be placed in a single group and subject to site root analysis.

Figure 7 also depicts analogous algorithms for identifying NamedTitle home pages (GA_t) and NamedURL home pages (GA_u) which differ from GA_p only in the choice of the marker and grouping inputs.

Extended site root analysis. In addition to the hierarchical structure inherent in the URLs, we noticed that a set of related pages in the IBM intranet often exhibited an implicit hierarchy in their titles. For example, the *Diversity Councils* home page is located within the website of the *Global Workforce Diversity* program which in turn is located under the main human resources website. The titles for these three pages are, respectively: "`You and IBM`", "`You and IBM | Global workforce diversity`", and "`You and IBM | Global workforce diversity | Initiatives | Diversity Councils`". Intuitively, web page authors use segments of the titles (the portions separated by "`|`") to indicate how the particular page fits within the overall organization of the web site.

To exploit this hierarchy, we developed another instance of the site root analysis algorithm as shown in Figure 8. In this case, the grouping input is the set of titles extracted from the entire corpus of intranet pages. The results of the three earlier global analysis algorithms (GA_p, GA_n, and GA_t) as well as the navigation panel local analysis algorithm (LA_n) are used as marker inputs. The grouping function is such that any two pages whose titles are hierarchically related (e.g., all three titles listed above) are placed within the same group. We refer to the output of this global analysis algorithm as the set of Rooted home pages.

2.3 Anchor Text Analysis

An important consideration in most implementations of Web search is the use of "*anchor text*". However, there is no single accepted approach and consequently there is a large body of work describing various implementations that exploit anchor text in different ways [5]. Our implementation to exploit anchor-text follows the two-phase strategy of local and global analysis. Following the template in Figure 1 anchor-text is treated as yet another "*navigational feature*"[6]. Further, we restrict the anchor-text string to the body of text within the "`<a href...>`" and "`</a>`" elements of a hyperlink. It is to this string that we apply the identical set of regular expression patterns used to identify candidate title home pages. These restrictions ensure that our exploitation of anchor-text meets the same stringent requirements as the other local analysis algorithms. At the end of local analysis, "*hyperlinked*" pages with anchor-text strings containing phrases such as "`John Smith's home page`" and "`Lenovo Intranet`" will be retained as candidate navigational pages. Note that analogous to candidate personal home pages it is possible to restrict attention by applying an additional dictionary filter, such that, in the above examples only "`John Smith's home page`" will be identified as a candidate personal home page.

In contrast to site root analysis — where the goal was consolidation from candidate navigational pages to root pages — the purpose of global analysis in anchor-text evaluation is to pick a particular feature value to associate with the candidate navigational page. Consider for example a candidate navigational page with four anchor-text features and associated values "`John R. Smith`", "`John R. Smith`", "`John Smith`" and "`Manager`"[7]. Global analysis operates by first creating groups of related features using an appropriate grouping function. A canonical feature value for the group with the largest number of elements is then determined. This is the feature value to be associated with the candidate navigational page. In the event where there is no group with the largest number of elements the candidate navigational page is discarded. In the above example using an "*exact match*" grouping function "`John R. Smith`" will be the feature value associated with the navigational page.

[6] Note that unlike title and URL anchor-text is an external navigational feature. External because it is not obtained from the page itself; instead it is extracted from pages "*hyperlinked*" to the page for which it is a navigational feature.

[7] Pointing out the obvious the first two feature values were extracted from anchor text "`John R. Smith's home page`", the third from "`John Smith's home page`" and the last from "`Manager's home page`".

3. INTRANET LOCAL SEARCH

As we mentioned in the introduction, one of the challenges of search in a large geographically distributed and diverse intranet is that of dealing with organization and location-specific terminologies, pages, and queries. In their study of the IBM intranet, the authors of [6] report that the correct answer to a query is often specific to a site, geographic location, or an organizational division, but the user often does not make this intent explicit in the query.

For example, a fairly popular query in the IBM intranet is "`bto`", an acronym that expands to Business Transformation Outsourcing. Using the analysis algorithms presented in the previous section, we were able to identify several home pages that were all associated with the feature value "`bto`" — including a BTO research home page, a BTO sales home page, and a BTO marketing home page. If a user who submits this query is primarily interested in sales figures and sales leads related to BTO customers, presenting him/her with the research home page is clearly not the best option. Ideally, if the profile of the user indicates that he/she belongs to the sales division (based on job description, job title, position in the organizational hierarchy, etc.), the sales home page for BTO must rank first in the search results.

In the same vein, there are several examples of navigational queries in the IBM intranet where the best result is a function of the geography of the user, i.e., the region or country where the user is located. For instance, there are about a dozen home pages that we identified as relevant for the query "`ip telephony`", one for each country where IBM has deployed the IP telephony program. More often than not, users who type in this query intend to reach the IP telephony home page for the country or region (Asia-Pacific, Latin America, etc.) in which they are located.

In general, given a search query with an associated user profile, our goal is to use certain attributes of the user profile, such as work location and job description, to further filter or rank the results from the navigational search index. Unlike on the Web, the task of associating a user profile with a search query is easily accomplished and poses fewer technical and privacy challenges[8]. However, two challenges remain to be addressed:

Task 1. Recognizing that a given navigational query is sensitive to a particular attribute of the user profile.

Task 2. Filtering or ranking the results from the navigational index given a particular attribute value.

In this section, we address a specific instance of the above problem, called "*intranet local search*",[9] where the attribute of interest is the "*geographic location*" of the user. Our solution consists of the following steps:

- *Geo-tagging*: a local analysis step in which each intranet page is individually analyzed and tagged with the names of one or more countries and regions.
- *Geo-sensitivity analysis*: a global analysis step in which the geography tags for all the pages with a given navigational feature value are examined to conclude whether queries matching that value are geography-sensitive
- *Geo-filtering*: a runtime filtering step in which the results for queries that are judged to be geography-sensitive are filtered to include only the pages from the geography where the user is located.

3.1 Geo-tagging

We have developed a local analysis algorithm for associating geographic tags with each page in the IBM intranet. Similar to the other analysis presented in Section 2, geography tagging consists of a feature extraction step followed by the application of regular expression and dictionary matches on the extracted features.

The set of features we extracted for each page consisted of the title, the URL, and the values associated with three specific HTML meta fields named "`Description`", "`Keywords`", and "`Country`". On each of these features, specially crafted regular expressions were applied to identify and extract names of countries, regions (e.g., Asia-Pacific, Americas, Eastern Europe, etc.), country codes, and region codes. To help with this process, we employed several generic as well as IBM-specific dictionaries such as a dictionary of country names, a dictionary of ISO country codes, a dictionary of geographic regions recognized within the IBM organization, and a dictionary of all known IBM sites along with their names and locations. The output of this step is a set of region and country names associated with each page in the corpus.

3.2 Geo-sensitivity Analysis

Geo-sensitivity analysis is required to ensure that filtering of results only happens for queries where such filtering makes sense. For example, if we examine all the pages associated with the navigational feature value "`ip telephony`", we see that the pages are distributed across a wide range of geographies. It is therefore a reasonable assumption that "`ip telephony`" is geography-sensitive and that results for this query submitted from, say IBM Italy, must include only the ip telephony home page for Italy or Europe. On the other hand, the query "`sam palmisano`" has nothing to do with geography and is always answered with the home page for Sam Palmisano, independent of the location from where the query is issued.

We are currently in the process of developing a sophisticated technique for classifying queries as geo-sensitive or otherwise. For the results presented in this paper, we employed the following simplistic metric: any query for which the results from the navigational index involved more than one country or region was presumed to be geo-sensitive.

3.3 Geo-filtering

On the IBM intranet, each query is associated with a particular user whose employee profile includes the country where the user works. Therefore, given a query that is judged to be geo-sensitive as per the metric described above, the the task of geo-filtering is merely one of removing all those result pages whose geography tag does not match with the geography of the corresponding user. Note that while the simple geo-sensitivity measure described above can be applied completely at query time by examining the result set from the navigational index, we expect more sophisticated geo-sensitivity analysis techniques to require offline pre-processing.

4. INDEXING NAVIGATIONAL PAGES

In this section, we address the task of building a "*navigational index*" that exploits the results of local and global analysis to answer navigational queries with significantly higher precision than a generic search index. There are two steps in this process: "*semantic term-variant generation*" and "*indexing*". We describe each of these steps below.

4.1 Semantic term-variant generation

Recall that at the end of the analysis described in Section 2 , we are left with multiple collections of navigational pages (Personal,

[8] This infrastructure is already available within the IBM intranet.

[9] This name is inspired by Web search services such as Google Local and Yahoo! Local

NamedTitle, NamedURL, etc). We refer to these collections as "*semantic buckets*".

Associated with each page in each bucket is a feature value — e.g., a person name, a phrase in the title, a segment of a URL, and so on. In the traditional indexing process, all of the terms in these features will be treated as regular text tokens and the resulting inverted index will only benefit from standard techniques such as stemming or stop word elimination. However, since each semantic bucket reflects the underlying analysis steps that was responsible for placing a particular page in that bucket, we can employ more sophisticated techniques when generating "*term-variants*" for these feature values.

Variant generator for person names. A rule based approach is ideal for navigational feature values that have very specific semantics that gets reflected in their structure. Among our examples, person names (i.e., the feature values associated with pages in the Personal bucket) fall into this category. Given a person name consisting of some subset of first name, middle name, last name, and nick name tokens, we have enumerated a set of rules for automatically generating all valid syntactic variants of that name. For instance, using our rules, we were able to generate 10 variants for the name "`Ching-Tien T. (Howard) Ho`" (e.g., "`howard ho`", "`ching-tien`", etc.).

Thus, an index that exploits these variants will be able to match a query "`ho, ching-tien`" against the feature "`Ching-Tien T. (Howard) Ho`" and return Howard Ho's home page from the navigational index.

Acronym-based variant generator. Recall our earlier observation (Section 2) about the prevalence of acronyms within the intranet and the fact that acronyms are a major fraction of the most common single word queries issued within IBM.

To take advantage of this observation, we first developed an acronym-finder by extending the techniques described in [18]. We ran this acronym finder over a corpus of $1.5M$ pages and were able to generate a dictionary containing approx. $27K$ pairs of unique acronyms and their expansions.

Using this dictionary, we implemented a variant generator that checks each navigational feature for the presence of an acronym or an expansion. Whenever an acronym (resp. expansion) is present in the feature, the corresponding expansion (resp. acronym) is used as a variant. Thus, the navigational feature "`Global Technology Services`" will result in a variant "`gts`" that will be indexed along with the original feature.

N-gram variant generator. Our default variant generator is a simple N-gram based approach that we use whenever the other two generators do not apply. Essentially, this generator treats all possible n-grams of a feature value as valid variants. Since typical keyword queries in the intranet are fairly short, we limit ourselves to n-grams for $n \leq 3$. Thus, the feature value "reimbursement of travel expenses" that we extracted from the title is associated with the following variants:

```
reimbursement
reimbursement travel
travel expenses
reimbursement travel expenses
```

4.2 Indexing

Once the appropriate variant generator has been applied to the feature values in each semantic bucket, the indexing process is straightforward. For each bucket, we build a corresponding inverted index in which the index terms associated with a page are derived exclusively from the navigational feature values and associated variants. None of the terms from the original text of the page are included. Thus the resulting inverted index is a pure "`navigational index`" that will provide answers only when user queries match navigational feature values or their variants.

5. RANKING ALGORITHM

After off line analysis (local, global and semantic variant generation) we are left with multiple indexes, each corresponding to one semantic bucket. During runtime, in response to a query, each index returns zero or more results and the task of the ranking algorithm is to merge these results into a single rank-ordered list. The focus of this paper is less on rank aggregation and merging and more on the discovery of semantic buckets. However we do implement a simple ranking algorithm based on the following statistical model.

Ranking based on expected precision. Let us assume that we have a set of queries Q and their relevance judgments. For every $q \in Q$ each of the indexes provides a ranked list. Thus to every index corresponds a dataset represented as a matrix where the rows are the ranks and the columns are the different queries q. The value of each cell is a pair of booleans indicating whether a result at this rank is provided for the given query and whether the answer is correct. Using this data we can estimate the probability of success for every index and for every rank, as $P_{t,r} = \frac{C_{t,r}}{N_{t,r}}$ where t subscripts the semantic bucket and r subscripts the rank, $C_{t,r}$ is the number of queries answered correctly at rank r by semantic bucket t, and $N_{t,r}$ is the number of queries answered at rank r by semantic bucket t. With sufficient data, a natural ranking order for the merged document list would be this estimate $P_{r,t}$. However in practice, with limited data, the estimated probabilities are not very reliable and consequently need to be smoothed. Moreover, since the final output is a ranked list any smoothing technique we use must account for the fact that probability of success for neighboring ranks should be highly correlated. This intuition is captured in the probability estimate calculated using the formula $P_{t,r} = \frac{\sum_{r-k}^{r+k} C_{t,r}}{\sum_{r-k}^{r+k} N_{t,r}}$ where k is a parameter which defines the window around which the probability of the current rank is computed. For our experiments we used a value of $k = 5$.

6. EXPERIMENTAL RESULTS

In this section we describe our experiments comparing the performance of the approaches presented in this paper (System X) against the existing system for IBM intranet search (W3). The goal of our experiments is two-fold:

Precision evaluation. Since our goal is precision, our main evaluation criterion is the popular Mean Reciprocal Rank (MRR).

Sensitivity to system parameters. Since we believe that intranets will be in a constant state of evolution a major goal in our undertaking is to build a system that can continuously be updated with new domain knowledge for both local and global analysis. It is therefore necessary to understand the overall effect to the system as parameters change.

6.1 Data set and queries

Our data set consists of a sample of about 5.5 million web pages taken from the IBM intranet. After offline processing we were left with approximately 55,000 pages - i.e., around 5 million pages were pruned away after local and global analysis.

The queries used in evaluation were selected from the top 10000 most frequent queries in the period June through August 2006. After removing duplicates we were left with 4759 distinct queries. From these we selected a subset of 346 queries for which we were able to get expert-defined gold standard consisting of 446 pages (some queries have more than one page that is the correct answer). All 446 results are present in our 5.5 million subset.

6.2 Evaluation criteria

A query is said to be "*answered*" by a system if the result list from that system is not empty. A query is said to be "*covered*" by a system if at least one result from that system matches the gold standard. Of the 346 queries, 321 were answered by System X, i.e. System X provides at least one result for each query. Since we are evaluating the precision of System X, all our results are reported on this set of 321 queries. The following criteria are used in our evaluation of the systems.

Mean Reciprocal Rank at 50 (M@50). MRR is the average, over all queries, of the reciprocal rank ($1/r$) of the best rank r of the correct results. The reciprocal rank is zero for queries without correct results. For computational expediency we only calculate M@50, where correct results beyond rank 50 are not considered.

Covered at 50 (C@50). The number of queries for which one of the correct answers is ranked in the top 50 in the result list.

Answered (A). The number of queries for which at least one answer is given (i.e. the result list is nonempty).

Success at 1 (S@1). The proportion of queries for which one of the correct answers was ranked first in the result list.

Success at 5 (S@5). The proportion of queries for which one of the correct answers was ranked in the top 5 in the result list.

6.3 Evaluation

Overall Performance of System X. Table 1 shows an overall comparison between System X and the existing system, W3. System X outperforms W3 on all the criteria. For example, out of the 179 queries System X answered with at least one correct result, 149 of which have at least one correct result ranked at 5 or better, giving $S@5 = 149/321 = 0.4642$. Note that System X provided a correct result in the top 50 hits in 56% of the time that it believed it contained the navigational page. Our data also show that rank 1 position of System X alone contains as many correct results as the first 10 positions of W3, while rank 1 and 2 positions of System X contain as many correct results as the first 48 positions of W3.

System	$M@50$	$C@50$	$S@1$	$S@5$	$S@50$
System X	0.3479	179	0.2679	0.4642	0.5576
W3	0.1799	110	0.1402	0.2274	0.3427

Table 1: Overall results using merged ranking

Effects of individual semantic buckets.. Table 2 shows the results for individual semantic buckets: Personal, NamedTitle, NamedURL and Rooted. The columns $C@50_X$ and A_X are the number of queries covered and answered by system X, respectively.

Although W3 results are not divided into the semantic buckets, we nevertheless calculates $M@50_{W3}$ for comparison purposes in the following way. First we note that W3 answers every query. For each query, we take the best ranked results from System X and find the semantic bucket(s) it belongs to. The $M@50_{W3}$ column is the MRR of a system that answers each query exactly in these buckets with the same result as that provided by W3.

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
Personal	0.5948	0.2649	36	58
NamedTitle	0.2226	0.1459	42	153
NamedURL	0.2907	0.1626	56	125
Rooted	0.1463	0.1637	99	290

Table 2: All keywords by semantic bucket

Effects of acronym dictionaries.. Table 2 showed that the Personal outperforms the other semantic buckets. Given that Personal was bootstrapped using an employee dictionary, we decided to isolate and measure the effect of using other dictionaries such as an acronym dictionary. Table 3 shows the results of additional semantic buckets formed by matching acronyms against NamedTitle, Rooted and NamedURL. The MRR of A-NamedTitle and A-NamedURL for System X are 65% and 23% higher than their non-acronym counterparts. The MRR of A-Rooted is smaller than that of Rooted, but adding this semantic bucket does not decrease the overall MRR, because the global ranking algorithm (Section 5) ensures that results with lower expected precisions do not get ahead of results with higher expected precisions.

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
A-NamedTitle	0.3619	0.2179	15	35
A-NamedURL	0.3568	0.1800	18	37
A-Rooted	0.1264	0.1711	17	58

Table 3: All keywords (Acronym)

Effects of keyword length.. The effects of semantic buckets are not uniform over queries of different length. Tables 4, 5 and 6 report the results on queries with 1, 2 and 3 keywords. It is interesting to note that NamedURL performs best on one keyword queries, Personal best for two keyword queries, while NamedTitle best for three keyword queries. The case for NamedURL is probably due to the fact that the URL names extracted contains a single path segment. The case for Personal is probably due to the fact that most personal name queries contain two keywords (first name and last name). The case for the NamedTitle is probably due to the fact that matching three keywords simultaneously against a string that is already identified as the name of a homepage is unlikely to be due to random accidents. Table 7 reports the results on queries with 1 keywords using acronyms. The MRR of all three semantic buckets improved over that of Table 4, due to many one-keyword queries involving an acronym that is matched in the title.

Effects of geography.. Table 1 shows that although System X answered 321 queries, only 179 contain correct answers in the top 50. Closer examination of results that did not contain correct answers show that many in fact contain geography sensitive homepages - those that should be considered homepages in a particular country but not globally. Performing geo-analysis on identified pages allows appropriate geography filtering based on the user geography.

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
Personal	0.0870	0.1674	2	23
NamedTitle	0.1664	0.1391	23	104
NamedURL	0.3018	0.1666	54	116
Rooted	0.1046	0.1322	55	175

Table 4: One keyword queries

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
Personal	0.9194	0.3029	30	31
NamedTitle	0.1838	0.1431	8	34
NamedURL	0.1481	0.1111	2	9
Rooted	0.1734	0.2125	32	95

Table 5: Two keyword queries

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
Personal	1.0	0.5625	2	2
NamedTitle	0.6250	0.2040	8	12
NamedURL	-	-	-	-
Rooted	0.3493	0.1653	11	19

Table 6: Three keyword queries

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
A-NamedTitle	0.3841	0.1366	10	23
A-NamedURL	0.3480	0.1787	13	25
A-Rooted	0.1624	0.1645	14	40

Table 7: One keyword queries (Acronym)

Table 8 shows the results for different user geographies. For example, out of the 321 queries that System X answered, 137 are pertinent to UK. Among these 137 queries, 99 are answered by System X with at least one answer pertinent to UK. The MRR of these answers is 0.4858. Comparing these results with Table 1, it is obvious that filtering the results by the user geography improves the MRR for all geography locations except US. The reason for this exception is likely due to the fact that, at least for IBM, the distinction between a GLOBAL page (i.e., a page with no particular geographical tag) and a US page is often difficult to ascertain.

Country	Region	$M@50$	$C@50_X$	A_X
UK	EMEA	0.4858	99	137
DE	EMEA	0.4879	104	144
US	AMERICAS	0.3384	164	293
CA	AMERICAS	0.4433	106	161
JP	ASIA-PACIFIC	0.484	101	138
AU	ASIA-PACIFIC	0.493	100	137
GLOBAL	GLOBAL	0.489	99	136

Table 8: Overall results using merged ranking and geography filtering (in the above, EMEA stands for Europe, Middle East, and Africa)

Results of anchor text analysis.. We also measured the performance of our implementation of the anchor text analysis algorithm described in Section 2.3. The results, labeled AnchorHP, are reported in Table 9. A comparison of the MRR values in Tables 9 and 1 reveals that even our fairly stringent implementation of anchor text performs worse than our other global analyses algorithms. However, we did notice that a significant fraction of the results from AnchorHP were in geographies that our current gold standard does not cover. It is quite likely that AnchorHP in conjunction with suitable geo-filtering might result in a fairly effective semantic bucket.

Semantic Bucket	$M@50_X$	$M@50_{W3}$	$C@50_X$	A_X
AnchorHP	0.1667	0.1401	7	36

Table 9: Anchor Text Analysis

6.4 Discussion

Several observations can be made on our experimental results:

- *A priori* identification of "*navigational pages*" boosts the MRR of the system significantly, compared with traditional information retrieval techniques.
- The separation of retrieved pages into multiple semantic buckets allowed us to produce a global ranking of the results that takes into account relative precision of various buckets.
- Domain dictionaries have significant value in improving the precision of many semantic buckets, and in creating new semantic buckets with higher precision.
- The relative strengths of semantic buckets are not uniform for different keyword lengths. This points to a way to further improve the overall MRR, by producing an overall rank of results based on keyword lengths in conjunction with semantic buckets.
- The geography analysis that consists of identifying page geography, user geography, query geo-sensitivity and results geo-filtering improves the precision significantly for geo-sensitive queries.

In general, our results appear to indicate the general benefit of "*stratification*" in constructing search engines. It allows improvements that are valid for particular aspect or subset of the results. A general ranking algorithm allows such improvements to be reflected in the overall results.

7. RELATED WORK

There are four broad areas of work that are relevant to the work presented in this paper. The following sections discuss related work in each of these areas.

Understanding search goals.. The classification of search queries into navigational, transactional, and informational was originally proposed in [1]. Several examples and scenarios for each class of queries in the context of enterprise search are described in [8] and a more recent analysis of user goals in Web search is presented in [17]. There has also been prior work in the use of techniques based on classification and user behavior for automatic user goal identification [11, 12, 15].

Transactional queries in the intranet has been investigated in [16, 10]. Analogous to our approach of pre-identifying and separately indexing navigational pages, the work presented in [16] describes a similar process for the class of transactional queries.

Web Genre Detection.. Automatic web genre identification (AWGI) is being recognized as a key factor for improving the quality of search results [4, 13]. For example, genre classification could allow users to sort search results according to their immediate interests [13] and could potentially improve navigational search as well.

Intranet search.. Upstill et. al. [19] investigate the use of evidence such as indegree, variants of PageRank, and URL-type, when identifying home pages on several test collections including an intranet data set. Their results indicate that of the three types of evidence investigated, reranking based on URL-type provided the maximum benefit.

The study on "workplace web search" by [6] established that several conventional ranking approaches that find favor in Web search are not effective discriminators when applied to intranet pages. Several aspects of their study that are particularly relevant to this paper have been mentioned in Section 1. The authors of [2] also elucidate the differences between search systems for the Web and those designed for enterprises.

Web Page Search.. There is a large body of work in the area of using structural information on a Web page (such as URL, anchor text, and title) to improve general Web search and link-based page classification [3, 10, 14]. Our work in this paper has shown how such structural information can also be used to identify navigational pages and thereby improve navigational search.

Finally, several of our local and global analysis algorithms are inspired by techniques (such as regular expression matching and dictionary matching) that are regularly used in "*information extraction*".

8. CONCLUSION AND FUTURE WORK

In this paper we have addressed the problem of answering navigational queries in an intranet setting. Learning from the experiences of previous studies [6] our system is designed explicitly to address several of the conditions that make search over intranets both non-trivial and different from search on the Web. In particular, our approach pre-identifies navigational pages during off line processing, classifies these pages into semantic buckets, and associates semantic term variants with each page prior to indexing. Our experiments over a corpus of 5.5 million pages from the IBM intranet demonstrated that this approach outperforms traditional WebIR ranking algorithms based on link analysis and static ranking. The ability of our approach to leverage domain dictionaries (such as acronyms and person names) was an important factor in improving the precision of our system. A particularly important result of our investigation was the powerful effect of incorporating geo-filtering to customize the results from our navigation index to the geographical location from which search requests are received.

Despite this success there are several areas that still need to be addressed. In Section 5 we have presented a simple, yet natural, rank-merge algorithm that was used to obtain the results presented in Table 1. Closer examination, however, reveals several inadequacies. For instance, there is a significant improvement due to the addition of acronym dictionary(cf. Section 6.3). However, merging the results from all semantic buckets in Table 2 and Table 3 increases the MRR by a mere 3%. Clearly we can do better and we intend to investigate other rank-merge algorithms. Another area that has the potential to provide dramatic improvements, such as those provided by geo-filtering, is ranking based on organizational hierarchies and job roles. Finally, we believe there is considerable gains to be obtained by going beyond semantic-variant generation and determining variants based on statistical analysis of the feature values associated for some semantic buckets.

9. REFERENCES

[1] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[2] A. Z. Broder and A. C. Ciccolo. Towards the next generation of enterprise search technology. *IBM SYSTEMS JOURNAL*, 43(3):451–454, 2004.

[3] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *SIGIR*, pages 250–257, 2001.

[4] A. Dillon and B. A. Gushrowski. Genres and the WEB: Is the personal home page the first uniquely digital genre? *Journal of the American Society of Information Science*, 51(2):202–205, 2000.

[5] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 459–460, New York, NY, USA, 2003. ACM Press.

[6] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson. Searching the workplace web. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 366–375, New York, NY, USA, 2003. ACM Press.

[7] M. Fontoura, E. J. Shekita, J. Y. Zien, S. Rajagopalan, and A. Neumann. High performance index build algorithms for intranet search engines. In *VLDB*, pages 1158–1169, 2004.

[8] D. Hawking. Challenges in enterprise search. In *15th. Australasian Database Conference*, 2004.

[9] Java regular expressions. http://java.sun.com/docs/books/tutorial/essential/regex/.

[10] I.-H. Kang. Transactional query identification in Web search. In *Asian Information Retrieval Symposium*, 2005.

[11] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *SIGIR*, pages 64–71, 2003.

[12] I.-H. Kang and G. C. Kim. Integration of multiple evidences based on a query type for web search. *Information Processing Management*, 40(3):459–478, 2004.

[13] B. Kessler, G. Numberg, and H. Schutze. Automatic detection of text genre. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, pages 32–38, 1997.

[14] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34, New York, NY, USA, 2002. ACM Press.

[15] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW05*, pages 391–400, 2005.

[16] Y. Li, R. Krishnamurthy, S. Vaithyanathan, and H.V.Jagadish. Getting work done on the web: Supporting transactional queries. In *SIGIR*, 2006.

[17] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA, 2004. ACM Press.

[18] A. Schwartz and M. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *Pacific Symposium on BioComputing*, 2003.

[19] T. Upstill, N. Craswell, and D. Hawking. Query-independent evidence in home page finding. *ACM Trans. Inf. Syst.*, 21(3):286–313, 2003.