Discovering Query Context using Concept Hierarchy

Mukesh Mohania
mkmukesh@in.ibm.com

1. Introduction

When we are searching information on Google (or any search engine), sometimes we wonder what keywords should be given to the search engine so that we get all the desired information that we are looking for. What happens next -- we keep changing the keywords until we get the desired information. That is, the onus of specifying appropriate set of keywords (called, query context) remains with the user, which is a limitation since the user might not be aware of the overall context at the point of submitting the query. This even becomes inevitable when the search query is submitted through the mobile device, particularly when the data source is not available all the time for searching or the bandwidth is limited. In such settings, the problem is how to derive the full context (as a set of keywords) of a query without sending the search query to the back-end data source. We propose to store and use the concept hierarchies at mobile device for discovering the query context. The advantage of discovering the query context is to further get all documents from enterprise content repositories and/or from external web which are highly relevant to the query results [1]. This provides a new way of integrating information [2] and also provides the user the ability to generate insights that would not be normally obtained by analyzing either type of information source (structured or unstructured) independently.

We discuss a method for discovering the context (in terms of a set of keywords) of a user query using the concept hierarchy at mobile device itself. The concept hierarchy [3] can be defined (either manually or semi-automatically) on the structured data. Concept hierarchy (could be data or application specific) provides a set of predefined hierarchical relationships that generalize lower layer (i.e., primitive data) information to high layer ones. For example, a set {tennis, rugby, hockey, football} can be generalized as “sports” at a high level concept. A concept hierarchy can be defined on one or on a set of attribute domains.

There are many applications where user would like to discover the broader context of his query derived from the concept hierarchies and also would like to get the relevant documents. For example, consider a stock-market information system. Such a system not only maintains the market statistics (structured data) but also the analyst advisories, risk- assessment reports, articles, related news, etc. (unstructured data). It would be nice if the stock trader, while querying the market statistics on, say, the fastest moving stock within a given sector at the moment, would also get the related advisories and reports. If he wants to trade on the stock, then depending upon the size of the trade, he gets the appropriate risk-assessment report. Note that these reports are available without his making an effort to hunt for them in the content repository, or on the web – *saving valuable time and effort*. Similarly, while browsing through an analyst report on a sector, it would be nice if the operator has access to the current statistics on the mentioned stocks without having to access them explicitly. Similar scenarios can be thought of in other domains also, e.g.:

- *Health:* Patient specific report and medical articles,
- *Manufacturing:* Defect statistics and engineering specifications,

- *Marketing:* Customer transaction history and marketing documents,
- *Travel:* Traveler itinerary and promotional flyers, travel advisories,
- *Management:* Employee records and status reports (details in Section **Error! Reference source not found.**)..

2. Research Issues

There are two main components of context discovery, namely,

- *Context Analyzer* analyses the input query and the concept hierarchy, and generates the context for the query (this is essentially as set of keywords obtained by navigating the concept hierarchy and any constrained neighborhood of the accessed database fragment).
- *Context Index* determines the set of documents in Content reposories most relevant to the context given as the input. Essentially, the input context is first mapped to a (small) set of relevant categories. Next, handles of the documents relevant to these categories are retrieved from a pre-computed index, optionally pruned, and output.

We now enumerate some of the research issues in discovering the context for mobile computing.

- Limitation: *Context is a set of keywords*. This is not very expressive. Can we do better than that? Including semantic information in the context appears to be an interesting issue for further exploration.
- Limitation: *The context of a search query is determined by concept hierarchy.* There exist other avenues that could be helpful in ascertaining the query context; such as the previous results retrieved and the query workload. If the user has provided a profile, that can be helpful too. Determining the query context from each of these dimensions, and consolidating the same appears to be an interesting research issue.
- Limitation: *Context of a search query is mapped to one or more categories; the set of documents associated with each such category is retrieved, and the union is returned as the set of documents relevant to the search query.* This strategy clearly suffers from a loss of precision. To improve precision, the final set of documents after the union can be further pruned based on the context, but it is not clear if it would be of significant help. More precise context-based document retrieval techniques need to be studied and/or developed.
- Limitation: *The documents and search query research are returned as unordered sets*. For usability reasons, efficient ranking algorithms to order the returned results (documents or database fragments) with respect to the input query context would be needed.
- Limitation: *Entire query results returned in addition to the documents on response to a query.* The query results need to be presented in a browse-able manner, or may even be presented as smart tags dynamically attached to the documents. This appears to be an interesting user interface research issue.

References

1. Roy, P., Mohania, M., Bamba, B., and Raman, S. Towards automatic association of relevant unstructured content with structured query results, CIKM (2005
2. Somani et al., Bringing together content and data management systems: Challenges and opportunities, *IBM Systems Journal, Vol. 41, No. 4, 2002*
3. Madria S, Mohania M, and Roddick, J, A Query Processing Model for Mobile Computing, FODO 1998