Best Bets: Thousands of Queries in Search of a Client

Giuseppe Attardi
Dipartimento di Informatica
Università di Pisa
via Buonarroti 2, I-56127 Pisa, Italy
+39 050 2212744
attardi@di.unipi.it

Andrea Esuli
Dipartimento di Informatica
Università di Pisa
via Buonarroti 2, I-56127 Pisa, Italy
+39 050 2212775
esuli@di.unipi.it

Maria Simi
Dipartimento di Informatica
Università di Pisa
via Buonarroti 2, I-56127 Pisa, Italy
+39 050 2212758
simi@di.unipi.it

ABSTRACT
A number of applications require selecting targets for specific contents on the basis of criteria defined by the contents providers rather than selecting documents in response to user queries, as in ordinary information retrieval. We present a class of retrieval systems, called *Best Bets*, that generalize Information Filtering and encompass a variety of applications including editorial suggestions, promotional campaigns and targeted advertising, such as Google AdWords™. We developed techniques for implementing Best Bets systems addressing performance issues for large scale deployment as efficient query search, incremental updates and dynamic ranking.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval – *information filtering, retrieval models, search process*.

General Terms
Algorithms, Performance, Experimentation, Theory.

Keywords
Information retrieval, information filtering, proactive content delivery, query, search.

1. INTRODUCTION
A wealth of new applications is being spurred on the Web by the wish of contents providers to assume a more active role in contents delivery: query-free search, advertising services (e.g. Google's AdWords and AdSense), editorial promotions, shopping advice (e.g. Amazon), recommending systems, matchmaking applications in e-commerce. An unusual feature of this class of applications is that they involve retrieval operations where the direction of search is reversed. In a traditional IR system, the user selects with a query the information he is looking for. In the mentioned applications, the (provider of) information selects among the users the most appropriate targets for specific contents. We call this class of applications *Best Bets* systems, since the contents chosen by a provider for a specific user is in a sense his *bet* on the fact that the user will find it interesting and often providers also bet against each other for user attention. Best Bets are similar to *Information Filtering* systems which involve matching contents against a collection of user profiles.

2. RETRIEVAL SYSTEMS
Retrieval systems can be characterized in terms of a general retrieval model and the retrieval functions they provide.

WWW 2004, May 17-22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

2.1 Retrieval models
A retrieval model provides an abstract description of the indexing process, the representations used for documents and queries, the matching process between them and the results' ranking criteria.

Definition 1. A *retrieval model* consists of a tuple ‹*D*, *Q*, *match*, *rank*› where:

1. *D* is a collection of documents
2. *Q* is query language
3. *match*: $Q \times D \rightarrow \{0, 1\}$, a query matching Boolean function
4. *rank*: $Q \times D \rightarrow [0, 1]$ a ranking function

Modern information retrieval systems, and in particular Web search engines, use a combination of the Boolean and vector space models: documents are selected according to Boolean combinations of term matching conditions (*match*) and the results are ordered according to a similarity measure as in the vector space model (*rank*). Matching may involve other conditions: for instance *proximity* or *phrase search* conditions.

2.2 Retrieval Functions
Each retrieval system or application provides a specific set of retrieval functions expressible in terms of the model. Here are some typical examples.

2.2.1 Document search
In Information Retrieval (IR), given a document collection *D*, the task is to retrieve all or the top *k* best ranking documents satisfying a given query q, i.e.

$$search(q, D) = \{ d \in D \mid match(q, d) \}$$

$$searchTop(k, q, D) = \text{top } k \text{ elements in } sort(search(q, D), rank(q, .))$$

where *sort*(*S*, *rank*(*q*, .)) sorts the documents in *S* according to *rank*(*q*,.), the ranking function partially applied to query *q*.

2.2.2 Query search
In Information Filtering (IF) the task is to match incoming documents against user profiles, expressed as queries. The difference between IR and IF is that "in filtering, an incoming stream of objects is compared to many profiles at the same time, rather than a single query being compared to a large, relatively static database" [3]. In IF the roles of documents and queries are swapped (Figure 1) and the task can be described as *query search*. Given a collection of queries *Q*, the goal is to find all queries *q* that match a given document *d*, i.e.

$$QuerySearch(d, Q) = \{ q \in Q \mid match(q, d) \}$$

2.3 Retrieval Function Support
In the abstract retrieval model, the direction of search is not accounted for. But the techniques devised for implementing a specific retrieval function are tailored to optimize one direction.

Document search, which retrieves documents from queries, exploits inverted lists, compressed posting lists, signature files and a number of query optimization techniques. These techniques are inappropriate for query search, where the task is to select queries that match a given document; in fact alternative techniques were proposed in [4] in the context of IF systems.

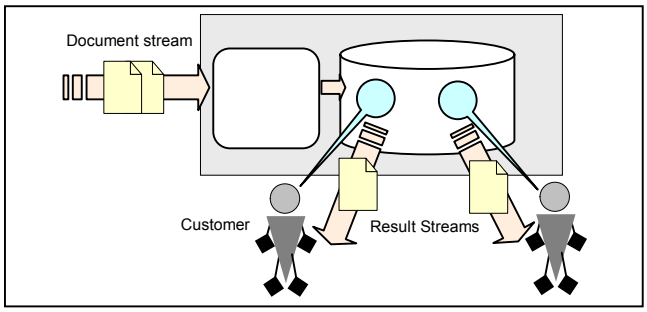

Figure 1. The IF model.

3. BEST BETS SEARCH

Besides Information Filtering, retrieval systems that invert the direction of search are useful in those cases where information providers wish an active role in selecting to which consumers their contents is delivered. As in IF, it is natural to express such selection criteria through a fully fledged query language.

In a Best Bets application a user/customer or his needs can be represented as a short-lived document that collects some aspects about him or about his activity context, e.g. input keywords to a search engine, navigation history or the document being browsed. Contents material is stored in a collection, where each item is paired with a query for identifying its intended target. When a user faces the system, all queries that match his description are selected and their associated contents are presented to the user ordered by a ranking measure (Figure 2).

Best Bets fit a retrieval model where the collection $D = Q \times C$ consists of pairs $\langle q, c \rangle$, where $q \in Q$ is a query and $c \in C$ is the associated document to be retrieved when the query matches a given input document d. Queries in Q represent target selection criteria for the items in C. For example, in an advertising application, d may be a user profile expressed as a list of keywords; Q contains queries expressing criteria for selecting potentially interested users for advertised products contained in C. The Best Bets task consists in retrieving all or the best k documents coupled to queries matching a given document d, i.e.

$$BestBets(d, D) = \{c \in C \mid \langle q, c \rangle \in QuerySearch(d, D)\}$$

$$BestBetsTop(k, d, D) = \text{top } k \text{ elements in } sort(BestBets(d, D), rank(., d))$$

Best Bets differ from IF since queries are associated to contents rather than to users, so documents have their own selection criteria and compete with each other through ranking, rather than being all delivered after a successful match.

In IF, as in IR, users determine with their queries (or profiles they control) the documents they want to receive; in Best Bets the producers devise queries to select customers for their documents. In IF profiles are matched with the document contents, so queries can discriminate only on what is present in them. Best Bets queries instead may contain criteria that are totally unrelated to the terms in the document: for instance a document about a pop singer might be paired to a query with the word "Mars", *betting* that interest in astronomy is more frequent in youngsters.

Ranking in Best Bets models the competition among providers by deciding which results appear in the top k positions, where k is usually a small number, as user attention is valuable. Hence ranking is crucial to determine if some contents will be delivered or not. Rank criteria must be fair and transparent to producers. They are typically based on parameters that vary dynamically and must be updated frequently, preserving an efficient computation.

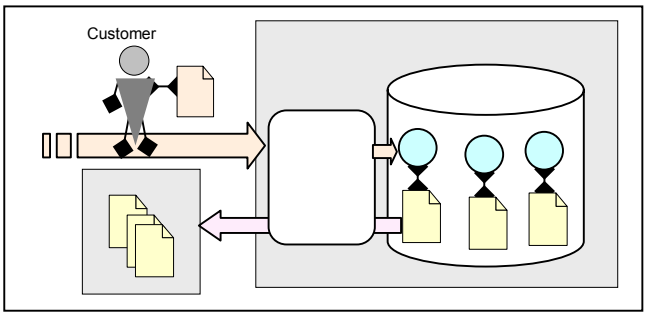

Figure 2. Best Bets model.

4. IMPLEMENTATION

We have developed [2] efficient implementation techniques for Best Bets applications and tested them in the deployment of an AdWords-like service [1]. The techniques are different from those described in [4] for IF systems as they aim to achieve:

Efficient query search: search time on complex query collections as fast as search on document collections of comparable size.

Incremental updates: updates to the query collection have immediate effects, without performance degradation.

Dynamic ranking: support for a ranking model based on continuously updated ranking parameters (after each search).

A two-level caching system is used to support real time updates of queries and ranking parameters. On a collection of one million queries running on a single PC, a steady performance of 180 searches per second was achieved stressing the system with 20 concurrent streams of queries and update requests. The cache helped sustain a rate of 200,000 updates, with less than 4% performance degradation. Dump of the updates from cache to disk takes about a minute, a reasonable time since a typical application is likely to take several hours to reach such volume of updates. Tests [2] using different collection sizes, cache sizes, and queries formulation (i.e. size, term dictionary and term frequency) showed near-linear performance scalability on these parameters, thus indicating the affordability of a large scale Best Bets service.

5. ACKNOWLEDGMENTS

KSolutions supported this research with a grant within the ClickWorld Project. We thank Antonio Cisternino for useful brainstorming sessions in earlier stages of this work.

6. REFERENCES

[1] Google, AdWords, https://adwords.google.com/.

[2] Attardi G., Esuli A., Simi M., Best Bets: Thousands queries in search of a client, TR-04-07, Università di Pisa, 2004.

[3] Belkin N. J., Croft W. B. Information Filtering and Information Retrieval: two sides of the same coin? *Communications of the ACM*, 35(12), 29–38, 1992.

[4] Yan T.W. and Garcia-Molina H., Index structure for Information Filtering under the Boolean Model. *ACM Transaction on Database Systems*, 19(2), 332–364, 1994.