Volume 2, No. 7, July 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

APPLICATION BASED UNDERSTANDING AND CLASSIFYING WEB QUIRES

Rajesh Kumar Ahirwar*, Mukesh Bhangre, and Rakesh Kumar Vishwakarma
Department of C.S.E. S.A.T.I., Vidisha M.P., India
ahirwarrajesh4@gmail.com
saimukesh411@gmail.com
rakesh26_kumar@yahoo.co.in

Abstract- Existing search services rely solely on a query's occurrence in the document collection to locate relevant documents. They typically do not perform any task or topic-based analysis of queries using other available resources, and do not leverage changes in user query patterns over time. In this paper provided within a set of techniques and metrics for performing temporal analysis on query logs. The metrics proposed for our log analysis are shown to be reasonable and informative, and can be used to detect changing trends and patterns in the query stream, thus providing valuable data to a search service. We continue with an algorithm for automatic topical classification of web queries. Results are presented showing that our classification approach can be successfully applied to a significant portion of the query stream, making it possible for search services to leverage it for improving search effectiveness and efficiency.

Keywords- Web mining, Classify Query Technique, Cluster.

INTRODUCTION

The information explosion of the WWW-era information age has made the field of Information Retrieval (IR) more critical than ever. There is a vast, growing expanse of data to search, and an expanding base of users with many diverse information needs. The ongoing struggle of information retrieval systems is to wade through this vast pile of data and satisfy users by presenting them with information that most adequately fits their needs. For many years, information retrieval research focused mainly on the problem of ad-hoc document retrieval, a topical search task that assumes all queries are meant to express a broad request for information on a topic identified by the query. This task is exemplified by the early TREC conferences, where the ad-hoc document retrieval track was prominent. In recent years, particularly since the popular advent of the World Wide Web and e-commerce, IR researchers have begun to expand their efforts to understand the nature of the information need that users express in their queries. The unprecedented growth of available data coupled with the vast number of available online activities has introduced a new wrinkle to the problem of search: it is now important to attempt to determine not only what the user is looking for, but also the task they are trying to accomplish and the method by which they would prefer to accomplish it. In addition, all users are not created equal; different users may use different terms to describe similar information needs; the concept of what is relevant to a user has only become more and more unclear as the web has matured and more inverse data have become available. Because of this, it is of key interest to search services to discover sets of identifying features that an information retrieval system can use to associate a specific user query with a broader information need.

All of these concerns fall into the general area of *query understanding*. The central idea is that there is more information present in a user query than simply the topic of focus, and that harnessing this information can lead to the development of more effective and efficient information retrieval systems. Existing search engines focus mainly on basic term-based techniques for general search, and do not attempt query understanding. This thesis addresses these shortcomings by presenting a pair of novel techniques for improving query understanding [1] and [2].

The ability to automatically classify queries by their intended topic and/or task of relevance is a key step forward in improving query understanding in IR systems. Our classification approach uses novel techniques along with tools and ideas from several disciplines in information science, including information retrieval, machine learning, data mining, and knowledge bases to achieve the most effective classification possible. We hypothesize that such a system is able to successfully classify a substantially larger portion of the query population than currently available techniques while requiring only a small amount of seed manual effort. We prove this hypothesis by evaluating our classifier over a large manually classified dataset, and we also compare our approach to the results of a recent competition on automatic query classification that was held at the KDD [3].

BACKGROUND

Examinations of search engine evaluation indicate that performance likely varies over time due to differences in query sets and collections. Although the change in collections over time has been studied (e.g., the growth of the web), analysis of users' queries has been primarily limited to the investigation of a small set of available query logs that provide a snapshot of their query stream over a fixed period of time. Existing query log analysis can be partitioned into large-scale log analysis, small-scale log analysis and some other applications of log analysis such as categorization and query clustering. A survey covering a great deal of relevant prior work in search studies can be found in a framework for static log analysis, but do not address analysis of changes in a query stream over time. Given that most search engines receive on the order of

between tens and hundreds of millions of queries a day, current and future log analysis efforts should use increasingly larger query sets to ensure that prior assumptions still hold. Previous studies measured overall aspects of users' queries from static web query logs. In the only large-scale study (all others involve only a few million queries), that users typically view only the top ten search results, and that they generally enter short queries [1].

Clustering

Cluster analysis has been successfully exploited in statistics, numerical analysis, machine learning and in other fields. The term "Clustering" denotes a wide range of methodologies for identifying hidden common structures in large sets of objects. A cluster is a group of objects whose members are more similar to each other than the members of other clusters. In this case, we say that intra-cluster similarity is high and inter-cluster similarity is low. Clustering methods are classified according to four aspects:

- ***The structure:*** This could be flat (there is no relationship between different clusters), hierarchical (clusters are organized in a tree), or overlapping (objects can be members of more than one cluster).
- ***The indexing unit:*** Documents are represented by means of a set of words, a so-called bag of words representation, or by means of sentences where the order of words is taken into account.
- ***The duration:*** The clustering is either carried out on top of a persistent collection of documents or on top of documents which exist for a very short period, like the set of search results given as an answer to a particular query submitted to a search engine. Several authors call this ephemeral clustering.
- ***The algorithm:*** It is used to generate the clusters, and could be divisive (starting from a set of objects and splitting them into subsets, possibly overlapping) or agglomerative (starting from individual objects and merging them into clusters).

Until a few years ago, persistent clustering was considered the "default" clustering technique, the cluster structure is generated only once, and cluster maintenance can be carried out at relatively infrequent intervals". The ephemeral clustering process organizes the documents in groups, which will survive just for the current session. Nowadays, ephemeral clustering is used by several search and meta-search engines to organize their results in fast brows-able groups. Surprisingly, ephemeral clustering has been less studied than persistent clustering in literature [2] and [6].

Prior Work in Query Log Analysis

Examinations of search engine evaluation indicate that performance likely varies over time due to differences in query sets and collections. Although the change in collections over time has been studied (e.g., the growth of the web), analysis of users' queries has been primarily limited to the investigation of a small set of available query logs that provide a snapshot of their query stream over a fixed period of time. Existing query log analysis can be partitioned into large-scale log analysis, small-scale log analysis and some other applications of log analysis such as categorization and query clustering. A survey covering is a great deal of relevant prior work in search studies. Framework for static log analysis, but do not address analysis of changes in a query stream over time [3] and [4].

Previous studies measured overall aspects of users' queries from static web query logs. No time-based or topic-based analysis of this query load was reported; it does not provide insight into how or when any usage or topical interest changes occur. Other studies examine the effect of advanced query operators on the search service coverage of Google™, MSN™, and AOL™, finding that in general, they had little effect. These overall statistics do not provide any insight into temporal changes in the query log, but do provide some insight into how people use search services [7].

It is well known that different users represent the same information need with different query terms, making query clustering attractive when examining groups of related queries. However, as traditional similarity measures are unsuitable for finding query-to-query similarity. It is incorporated click-through to cluster users' queries. In evaluating their system, they analyzed a random subset of 20,000 queries from a single month of their approximately 1-million queries-per-week traffic. Previously found that the most popular 22.5% queries represent only 400 clusters of queries using differing sets of query terms.

A more recent study used temporal correlation to find sets of similar queries, suggesting that queries with similar frequency patterns are likely to be related. They defined a formal metric for temporal similarity between queries and used it to mine sets of related queries from a six-month MSN™ search log. The presented results are largely anecdotal, but suggest a promising technique provided that noisy, unrelated queries can be adequately handled [5].

Many web search services have begun to offer views of the most popular and/or changing (becoming drastically more or less popular) queries: AOL Member Trends, Yahoo - Buzz Index, Lycos - The Lycos 50 with Aaron Schatz, Google Zeitgeist, AltaVista - Top Queries, Ask Jeeves, Fast (All The Web). These views necessarily incorporate a temporal aspect, often showing popular queries for the current time period and those that are consistently popular. Some also break down popularity by topical categories. Systems seeking to display changing queries must address the issue of relative versus absolute change in a query's frequency to find queries whose change is interesting, not simply a query that went from frequency one to two (a 200% jump), or one that went from 10,000 to 11,000 (an absolute change of 1,000) [1] and [8].

PROPOSED TECHNIQUES

Automatic Classification of Web Queries

Accurate topical classification of user queries allows for increased effectiveness, efficiency, and revenue potential in general-purpose web search systems. Such classification becomes critical if the system is to return results not just from a general web collection but from topic-specific back-end databases as well. Successful query classification poses a challenging problem, as web queries are very short, typically providing few features. This feature sparseness, coupled with the dynamic nature of the query stream and the constantly changing vocabulary of the average user hinders traditional methods of text classification. To mitigate this, we propose a multi-part approach: we combine the benefits of matching against a list of manually classified queries and

supervised learning of classifiers with a novel application of selectional preferences. We mine large, unlabeled web query logs for rules based on strong selectional preferences and use these rules to capture latent expressions in the query stream associated with topical categories. This technique allows our classifier to remain abreast of changes in the query stream by updating the rules over time using fresh logs. Combining these techniques allows us to classify a substantially larger proportion of queries than any individual technique. We evaluate our approach using a large sample of queries from a real web query stream.

Understanding the topical sense of user queries is a problem at the heart of web search. Successfully mapping incoming general user queries to topical categories, particularly those for which the search engine has domain-specific knowledge, can bring improvements in both the efficiency and the effectiveness of general web search.

Proposed Classification Approaches

Prior efforts in classifying general web queries have included both manual and automatic techniques. In this section, we describe the manual and automatic classification approaches that collectively form our framework and explain our motivations for using each approach. We also introduce a new rule-based automatic classification technique based on an application of computational linguistics for identifying selectional preferences by mining a very large unlabeled query log. We demonstrate that a combination of these approaches allows us to develop an automatic web query classification system that covers a large portion of the query stream with a reasonable degree of precision.

Exact-Match Using Labeled Data

The simplest approach to query classification is looking the query up in a database of manually classified queries. This is not quite as crazy as it sounds. At any given time, certain queries (the reader can anticipate some of these) are much more popular than others. By combining manual classification of these queries with acquiring large databases of proper nouns in certain categories (personal names, products, geographic locations, etc.), non-trivial coverage of the query stream can be achieved.

N-Gram Match Using Labeled Data

In addition to the problems with manual classification discussed above, there is the significant issue of query ambiguity. Many queries are ambiguous, making assessor disagreements inevitable. Bulk loading of large lists of entities contributes to this problem through mismatches in category definitions and introduction of possible, but low frequency, interpretations for queries. To mitigate this, we incorporate a technique similar to exact-matching, where instead of performing exact-match lookups, we n-gram incoming queries first, and make the assumption that if an n-gram of a query appears in a given category, the query is associated with that category. We experimented with both word-based and character-based n-grams, and found word-based 4-grams to be the most effectiveness for this approach. Although using the n-gram approach alongside exact match lookups gives some improvement, it is immediately clear that performing lookups into manual classifications is at best a high precision, low recall approach, insufficient as a standalone technique if the end goal is to develop a high-recall classification system for general web queries. As such, we use these techniques as components in our combined approach, taking advantage of their precision while relying on other approaches to enhance classification recall.

Supervised Machine Learning

A natural next step is to leverage the labeled queries from the exact match system through supervised learning. The idea is to train a classifier on the manual classifications with the goal of uncovering features that enable novel queries to be classified with respect to the categories. A challenge for this approach is that web queries are short, averaging between 2 and 3 terms per query. This leaves a learner with very few features per example.

To examine the effectiveness of the supervised learning approach we trained a perceptron for each category from the exact match system, using all queries in a given category as positive examples and all queries not in that category as negative examples. To realistically evaluate this approach we developed a test collection that is representative of the general query stream. To determine the number of queries required to achieve a representative sample of our query stream, we calculated the necessary sample size in queries:

$ss = (Z^2) (p) (1-p)/c^2$

Equation Representative Sample Size Formula

Where:

Z = Z value (e.g. 1.96 for 95% confidence)

p = percentage picking a choice, expressed as decimal (0.5 used here)

c = confidence interval, expressed as decimal (e.g., .04 = +/- 4)

Selectional preferences can be used for disambiguation and semantic interpretation. If *x* strongly favors *y*'s that belong to class *u*, then *u* is a good prediction for the class of an ambiguous, or previously unknown, *y* in that context. Indeed, many studies have evaluated the quality of learned selection preferences by measuring the accuracy with which they can be used for disambiguation.

To take advantage of this disambiguation effect to classify queries, we use a large log of unlabeled queries and do the following:

1. Convert queries in an unlabeled query log to a set of head-tail (*x; y*) pairs.
2. Convert the (*x; y*) pairs to weighted (*x; u*) pairs (where u represents a category), discarding *y*'s for which we have no semantic information. Do this in both the forward (*x; u*) and backward (*u; x*) directions.
3. Mine the weighted pairs to find lexemes that prefer to be followed or preceded by lexemes in certain categories (preferences).
4. Use the mined preferences to assign test queries to semantic classes.

Step 1 is straightforward for queries, *vw* of length 2. We have only two possible \syntactic" relationships: the first token providing context for the second, or the second providing context for the first. These produce the forward pair (*-v;w*) and the backward pair (*w ; v*), where the underscore indicates matching at the front or the back of the query, respectively. We keep pairs of the two types separate, and call selectional preferences mined from (*v;w*) pairs forward preferences, and those from (*w-; v*) pairs backward

preferences. It is clear that this technique is not applicable to single-term queries, as there is nothing present to provide context for a single term (it should be noted that approximately 15% of all queries are composed of a single term).

If all two token queries were simple noun phrases (a modifier followed by a noun), then forward preferences would capture the degree to which a particular modifier (noun or adjective) constrained the semantics of the head noun. Backward preferences would capture the reverse. In practice, two token queries can arise from a variety of other syntactic sources: verb-object pairs, single words spelled with two tokens, non-compositional compounds, proper names, etc. A user may also intend the query as a pair of single words in no particular syntactic relation. Typographical errors and other anomalies are also possible. Thus, our forward and backward relations inevitably have a murky interpretation.

Longer queries have more structural possibilities. Rather than attempting to parse them, we derived from a query *abc. . . uvw* all pairs corresponding to binary segmentations, i.e.:

(*a; bc: : :w*); (*ab; c: : :vw*); *: : :*(*abc: : :v;w*)
and
(*bc: : :w; a*); (*c: : :w; ab*); *: : :*(*w; abc: : :v*)

For any given query, most of these pairs get screened out in Step 2.

In Step 2, we replace each pair (*x; y*) with one or more pairs (*x; u*), where *u* is a thesaurus class. Pairs where *y* is not present in the thesaurus are discarded, and pairs where *y* is ambiguous yield multiple fractionally weighted pairs. Our thesaurus" is simply our database of manually classified queries, with each query interpreted as a single (often multi-token) lexical item. The possible semantic classes for a lexical item are simply the set of categories it appears under as a query.

In Step 3, we compute $S(x)$ for each x, as well as the MLE of $P(u|x)$ for each ($x; u$) pair seen in the data. We then screen out pairs where $S(x) < 0{:}5$, a relatively low threshold on selectional preference strength determined by initial tests on our validation set. From each remaining pair we form a rule $[x \rightarrow u : P(u|x)]$, which is interpreted as saying that a query matching x gets a minimum score of $P(u|x)$ for category u. If ($x; u$) was a forward pair (i.e. x is - v"), we require x to match a prefix of the query for the rule to apply, while if ($x; u$) is a backward pair we require x to match a suffix of the rule to apply.

Finally, in Step 4 we use selectional preferences to classify test queries. We attempt to match each forward selectional preference against the initial tokens of the query, and each backward preference against the final tokens of the query. We give the query a score for each category u corresponding to the maximum $P(u|x)$ value of any rule that matches it. We then compare the maximum $P(u|x)$ values for a query against a threshold tuned to optimize classification effectiveness and assign the query to all u's with values that exceed the threshold. Tuning is necessary since the $P(u|x)$ values are estimates of the probability that a subpart of the query would be viewed as belonging to a category (with categories considered as mutually exclusive), not estimates of the probability that the whole query would be viewed as belonging to that category (with categories considered as overlapping).

RESULT

Determining the topic of unrestricted web queries is an important problem in modern web search. This is increasingly true as modern search services continue to move forward from basic text search into the realm of searching large numbers of specialized backend databases. We developed a system for automatic web query classification that combines a small manual classification with techniques from machine learning and computational linguistics. This combined approach can correctly classify up to 30% of queries while still maintaining reasonable precision, and outperforms the recall of the best single approach by almost 60%.

CONCLUSIONS AND FUTURE WORK

Our combined approach is able to leverage the specific strengths of each individual approach to classify a much larger portion of the query stream than would be possible using any of the individual methods alone. Moreover, by leveraging the unlabeled data contained in user query logs, we have created a classification system that is automatically robust to changes in the query stream, for as the users' change their queries, a system that relies on query logs for information will be able to adapt to them immediately and automatically. Because of this, we can minimize the need for periodically labeling new training data to keep up with changing trends in the query stream over time, and we are not forced to rely on computationally prohibitive forms of external information to maintain our classification effectiveness.

There are several potential areas for future work, including: Expanding the existing manual classification with queries that are classified by the framework, and sub setting the manually classified queries by linguistic properties. For improving the linguistic properties of our selection preference algorithm by more explicit handling of semantic and structural ambiguity, and improving its machine learning properties by incorporating ideas from traditional rule learning. Examine the effectiveness of each approach on popular queries versus rare queries (queries in the tail of the stream). Examining specific topical categories where one approach outperforms another and using this information to create more sophisticated combined classification techniques.

REFERENCES

[1] Beitzel, S. M., E. C. Jensen, D. D. Lewis, A. Chowdhury, A. Kolcz, and O. Frieder. “Improving automatic query classification via semi-supervised learning." In The Fifth IEEE International Conference on Data Mining, pages 42{49,Houston, TX, 2005. IEEE Computer Society Press.

[2] Grossman, D. A., S. M. Beitzel, E. C. Jensen, and O. Frieder. “The IIT intranet mediator: Bringing data together on a corporate intranet." IEEE IT Professional, 4(1):49{54, 2002.

[3] Spink, A., B. J. Jansen, D. Wolfram, and T. Saracevic. \From e-sex to ecommerce: Web search changes." IEEE Computer, 35(3):107{109, 2002.

[4] Xie, Y. and D. O'Hallaron. “Locality in search engine queries and its implications for caching." In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, NY, 2002. IEEE.

[5] Beitzel, S. M., E. C. Jensen, A. Chowdhury, D. Grossman, O. Frieder, and N. Goharian. “On fusion of effective retrieval strategies in the same information retrieval system." Journal of the American Society for Information Science and Technology, 55(10):859{868, 2004.

[6] Chakrabarti, K., S. Chaudhuri, and S.-w. Hwang. “Automatic categorization of query results." In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pages 755{766, Paris, France, 2004. ACM Press. http://doi.acm.org/10.1145/1007568.1007653.

[7] Chien, S. and N. Immorlica. “Semantic similarity between search engine queries using temporal correlation." In Proceedings of the 14th International Conference on the World Wide Web (WWW), pages 2{11, Chiba, Japan, 2005. ACM Press. http://doi.acm.org/10.1145/1060745.1060752.

[8] Cronen-Townsend, S., Y. Zhou, and W. B. Croft. Predicting query performance. "In Proceedings of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pages 299{306, Tampere, Finland, 2002, ACM Press.

