Unity: Relevance Feedback using User Query Logs

Jignashu Parikh
Yahoo! SDC
Bangalore, India
jignashu@yahoo-inc.com

Shyam Kapur*
Adchemy Inc.
Palo Alto CA 94306
shyam_kapur@yahoo.com

ABSTRACT

The exponential growth of the Web and the increasing ability of web search engines to index data have led to a problem of plenty. The number of results returned per query is typically in the order of millions of documents for many common queries. Although there is the benefit of added coverage for every query, the problem of ranking these documents and giving the best results gets worse. The problem is even more difficult in case of temporal and ambiguous queries. We try to address this problem using feedback from user query logs. We leverage a technology called *Units* for generating query refinements which are shown as *Also try* queries on Yahoo! Search. We consider these refinements as sub-concepts which help define user intent and use them to improve search relevance. The results obtained via live testing on Yahoo! Search are encouraging.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval- *Query formulation, Relevance feedback*

General Terms

Algorithms, Experimentation

Keywords

Information Retrieval, Relevance Feedback, Query Expansion, Query Refinements, Search Re-ranking, Search Engines

1. INTRODUCTION

The exponential growth of the Web and the increasing ability of web search engines to index data have led to a problem of plenty. Yahoo! Search [3], for instance, recently announced an index size of 20 Billion documents. While the number of documents is growing, the user search query patterns remain largely unchanged. Recent studies about search engine queries indicate that most users still use two word queries [1]. The search engines' indexes typically have a very large number of documents for most common 1-2 keyword queries. Let us consider the desired relevance for a few such queries.

Python is an ambiguous query with the dominant senses of a *snake* and a *programming language*. Such ambiguous queries are amongst the biggest challenges for search engines. It can be argued that link patterns on the Web give an indication of how dominant a particular sense is. However, we feel that linking is biased and an indication of this are the search results of Google [4] for the query *python* which shows most results in the top ten belonging to the programming language sense. Comparing these to *Also try* queries, which are an indication of the most common refinements associated with the given query and are available under the search box on Yahoo! Search, one finds a discrepancy. The first five refinements are: *monty python, ball python, colt python, burmese python, python snake.* A possible reason for this discrepancy is that web page creators are more tech savvy (in terms of linking documents, creating content) than average users. Thus, the links that are created online may not be the best measure for disambiguation of an ambiguous query. In addition to the above, there are queries like *Britney Spears*, which is often an *in-news* query, and queries like *Turkey,* which can be a *seasonal/temporal* query. These issues of ambiguity, variety, freshness and temporality are well known in the IR community but are not fully addressed by major web search engines. In this paper, we propose a solution to address these issues in a generic way using user query logs. The central idea is to use search refinements as a source for establishing a global truth (sense) for a given query and tune our search results in various ways including query expansion, interleaving results, and re-ranking of results. We generate search refinements using a technology called *Units* (See also: [1]). *Units* are defined as the atomic constituents or "concepts" in user queries. They are generated from the search query logs through an automated statistical approach.

2. UNITS

User queries are made up of one or more words. Some examples are *hawaii*, *new york city*, and *new york city law enforcement*. We believe that human beings do not naturally think in terms of queries but rather in terms of *natural concepts*. Hence, *hawaii* and *new york city* are different queries as measured by number of words but they are both made up of one concept each. *new york city law enforcement* is different from them as it is made up of two distinct concepts *new york city* and *law enforcement*.

Units are generated using an iterative statistical approach using canonicalized query logs. The Unit generation algorithm takes as its input a consolidated query file, which has distinct queries and their frequencies over a specific time interval. In the first iteration, all single words in all queries are considered as *units* or concepts and their frequencies are stored. In subsequent iterations, every query is tokenized into the existing set of units. Co-occurring units are merged to form possible units and are stored to record their total co-occurrence frequencies over the entire query file. At the end of the iteration, possible units are validated to determine if they are actual units (based on mutual information and other statistical measures).

One of the important applications of Units is the generation of query refinements. Query refinements are determined from what are called *extensions* and *associations*. An *extension* of a unit is a larger unit that contains all the words in the first unit. An *association* of a unit is another unit with which the first unit appears often in queries. We applied the Units and refinements generation processes on Yahoo! Search query logs. The resulting refinements have been tested and the suggestions are used as one of the inputs in the *Also try* feature provided on the Yahoo! Search results page since the Spring of 2003.

* This work was done while the author was working at Yahoo! Inc.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
ACM 1-59593-369-7/06/0008.

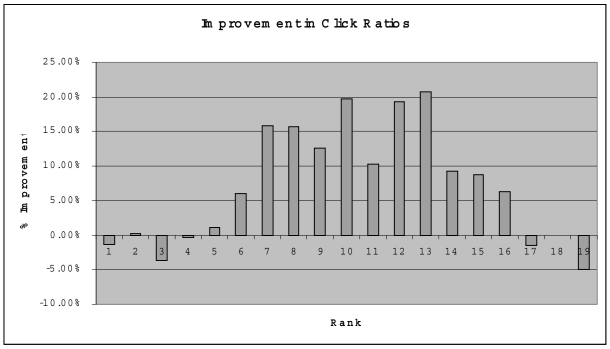

Figure 1: Percentage Improvement due to Unity

3. UNITY

The Unity system leverages the Units technology and the refinements generated using the Units technology for enhancing search results. Refinements provide an important insight into what sub-concepts users have used to clarify the exact intent of their original queries. Consider the example query *party.* While the search results of both Yahoo! Search and Google show sites related to political parties as their top results, the top five refinements for this query during December, 2005 wer*e party poker, new years eve party, party city, party games, party supplies.*This clearly shows the dominant user intents for the query *party*. We have explored the following algorithms for improving relevance using query refinements:

1. *Implicit Re-ranking:* This algorithm uses the search results from the original query and its best *K* (typically K=5) refinements. We then merge (and re-rank) the results based on occurrences of the first *M* (typically M=20) related searches in the title and abstract of the search results. A variant of the algorithm uses the actual documents corresponding to the various search results, which can be expensive but more accurate.
2. *Explicit and Implicit Interleaving:* This algorithm doesn't disturb the original search engine's ranking but introduces the results of the refinements at regular pre-defined positions. For example, one result each for the first N refinements can be inserted at every fifth position in the original result set. This method introduces relevant results *missing* from the top results of the search engine and also adds variety to the result set. The interleaved query results can be *explicitly* marked as coming from a query refinement or can be left unlabelled.
3. *Click based Replacement:* The previous algorithm removes the last few results from specific, usually low, positions on the first search results page. Instead, we can replace those results selectively by choosing results which have *less than expected* click values based on previous weeks' click logs. This is an effective and low-risk algorithm as it only replaces results which the users don't relate to.

While we implemented and tested *implicit re*-ranking, so far we have only prototyped *Implicit/Explicit interleaving* and are planning to build *Click based replacement*.

4. EVALUATION

We evaluated *Implicit Re-ranking* live on Yahoo! Search. We had used only the title and abstract of the results for calculating relevance scores as per the *implicit re-ranking algorithm*. Thus, we assumed that clicks based on the title and abstract shown on search results page would indicate that the users found the suggested results relevant. This test was conducted using a benchmark of 10000 queries selected randomly from a subset of queries which were frequently refined by users. The top 20 results for each query were then generated using the *Implicit re-ranking algorithm*. The re-ranking was done such that there was rarely a change in the top 5 results for every query. The test was conducted for a week where a bucket of 2% of Yahoo! Search users was targeted with our results and there was a control bucket of 2% users who were shown the original search results. The users in both buckets were chosen randomly. We were interested in finding out the click distribution for the top 20 ranks coming from each of the buckets.

Figure 1 shows the improvement in percentage clicks received by Unity as compared to default Yahoo! Search results at various ranks from 1-20. The results show a very positive indication about the relevance of Unity results. The graph shows significant improvements with Unity results below rank 5 after which most of Unity's re-ranking came into play. It should be noted that we get this improvement even after introducing the results beyond the first five results, which tend to get most of the user attention. Users had to scroll or go to the next page (if their number of results on first page is set to 10) to see the newly introduced results. We plan to conduct more tests with introduction of more Unity results starting higher up than rank 5. We would also like to integrate the system more closely with the search engine and evaluate the impact.

5. CONCLUSION

The paper described a novel method for relevance feedback using user query logs. The proposed algorithms make use of refinement queries as an indication of concepts closely related to the given query's original concept. We make use of a technology called Units for generation of query refinements. Our tests show promising improvement over Yahoo! Search's original results in terms of click ratios. As mentioned in the previous section, we plan to conduct further aggressive tests starting the re-ranking from ranks higher than 5 including 1st and 2nd ranks. We would also like to evaluate the other two strategies, namely, Implicit/Explicit interleaving and Click based replacement further. It should be noted that the proposed techniques can be applied in the area of any vertical search products like news, blogs, shopping, images, video, etc. We can make use of query refinements from the vertical or web search or a merge of both for this purpose. Similarly, instead of considering refinements generated from weekly logs, we can consider refinements generated per day or even per hour to identify any change in user interests as early as possible. Towards an end goal, we would like to investigate how we can pursue personalization of search results using user specific search query refinements.

6. REFERENCES

[1] Pavlov, D., Balasubramanyan, *et al*. Document preprocessing for naive Bayes classification and clustering with mixture of multinomials. In *Proceedings of KDD* pp. 829-834*, 2004.*

[2] OneStat.com: *Most people use 2 word phrases in search engines*, August 2004. http://www.onestat.com/html/aboutus_pressbox27.html

[3] Yahoo! Search: http://search.yahoo.com

[4] Google Search: http://www.google.com