

# Leveraging Query Associations in Federated Search

Aditya Pal  
Amazon  
Bangalore, India  
adityap@amazon.com

Jaya Kawale  
Yahoo R & D  
Bangalore, India  
jaya@yahoo-inc.com

## ABSTRACT

There has been a quantum increase in content published over the world wide web. Search Engines have done a good job in crawling and indexing the web. Search engines generally return results from different context or categories and user modifies his queries to get a more precise list of results. Also, a user might need to query several search engine sources (e.g. videos, photos, news, answers etc) before he can find a satisfactory answer to what he is looking for and his modifications for same context in each search engine might vary.

We present SAFE (Search Aggregation and Federation Engine) a multiple content source search engine that allows users to simultaneously search across multiple search engines. It extends the idea of federation in search and proposes unique ways to capture query modifications that is one of the quintessential technique used by users to get more relevant results. We use the query modifications to enrich the search experience of a user in a federated search environment by making the search page more content rich, relevant and personalized.

## 1. INTRODUCTION

*Meta search* is the simultaneous search of multiple search engines and linking them together as one virtual search system. *Meta search* (also known as *federated search*<sup>1</sup>) consists of following steps (1) *transforming a query* and firing it to several search engines with appropriate syntax, (2) *merging the results* collected from the individual search engine to a single ranked list or keeping them separately according to their source, (3) *presenting the results* in a succinct format to the user [3].

*Meta search engine* combines results from several heterogeneous search engines. The mixing of results from individual search engines to a single ranked list can be effective if all the results are of same *media type* (either textual or videos or images) [30, 31]. Even for results of the same *media type* but with different context,

<sup>1</sup>Federated search and Meta search are used interchangeably in this paper.

mixing to form a single ranked list might not be desirable. For Example, mixing news results with Q&A results and web results can lead to clumsy user interface and user can lose focus due to context switching. For a search system which combines results from search engines with *heterogeneous media types* or *different contextual information*, a *modular approach* is more preferred as in [13, 11]. Each *module* represents results from one search engine and the results are not mixed across modules.

Search engines normally return results from multiple categories or contexts and users have to modify their queries to get more precise results. Query modifications by a user can differ in different search engines due to the difference in contextual information search engine returns. For Example, a user with intent of *visiting Mauritius* can search for *best time to visit Mauritius* in *Yahoo Answers* and *beach pictures of Mauritius* in *Yahoo Image Search*. Additionally, due to difference in relevance of search engine results [29, 24], a user searching for the same concept in two different search engines might form different query modifications in both of them to get more precise results. Even, if the user forms same query modifications, he might end up re-modifying the query modification in one search engine, whereas he might be satisfied with the results in other search engine.

In this paper, we present a meta-search engine named SAFE (Search Aggregation and Federation Engine) which uses *Yahoo Developer Network's* [15] public apis to get results from various search engine sources such as *Yahoo Image Search* [16], *Yahoo Web Search* [19], *Yahoo News Search* [18], *Yahoo Answers* [14], *Flickr* [4], *Yahoo Videos* [20], *Yahoo Maps* [17], etc and presents them in a modular user interface similar to [13, 11] to the user. We present a novel technique to capture a query and it's modifications in different modules (also referred as *Query Association* or simply *Association* in this paper). Based on the *Query Associations*, we propose query refinement scheme and module ranking scheme for modules in SAFE. We also mention some useful features implemented in SAFE that improves federated search experience of the user.

## 2. RELATED WORK

Meta-search engines have been around for quite some time. Clusty [1] one of the earliest such engines uses Ask, Live, Gigablast, etc. to present a unified result list to the users. In addition to that it also clusters the results into several categories and allows users to navigate into these categories. Dogpile [2] and Mamma [6] are other meta search engines available which present results as single list to the users. Mixing results in federated search, based on search engine result effectiveness can improve the federated search engine effectiveness [31].

On the other hand Alpha Yahoo! [13] and Search Mash [11] take a *modular approach* of showing separate lists of the individual search engine results to the user. In their *user interface*, they keep one module as main module and other modules appear collapsed by default or appear at one side. User can click the module in order to view results from it indicating his current interest to the meta-search engine.

The article *Meta-search: More heads better than one ?* [7] points out several advantages of meta-search engines, particularly those who mix results to form a single ranked list, like, consensus on results increase the confidence factor in the results. However major drawbacks of meta-search approaches used so far is that they still haven't tapped upon the huge advancements in web search technology.

Search engines normally return results from multiple categories or contexts for a query and users have to modify their queries to get more precise results. One approach search engines follow to understand web queries is to subsequently expand them using the query reformulations performed on a query in a session. There is a lot of work in this direction as mentioned in [21, 28, 22, 27]. However there is no similar approach to understand queries in a meta-search search context where there are multiple distinctive content sources. Our approach tries to tie query reformulation in such contexts by forming a notion of query associations.

### 3. QUERY ASSOCIATION

Search Engines have done a great job in crawling and indexing web pages from all the accessible parts of the Internet [10]. When a user searches for a query term, they return results from different relevant categories or context to the user. User has to modify their queries, based on their context, in order to get more precise search results (i.e. results from a particular category or context). Research has been done to personalize the search for the user based on his previous searches [26, 25]. Google offers personalized search engine which provides more relevant search results based on user's previous search history [5]. Personalization certainly helps in retrieving more relevant results but it cannot adapt to changes in user's interest and likeness. Personalization cannot unlock the context transitions that go in user's mind leading users to switch context. Query modification is the fastest way for a user to get the most precise results.

Subsequent user queries (within a short time interval) are inter-linked and the users transition from one query to another form a trail. A user starts searching with some concept and later refines his concept based on the results he gets. This *trail of modification* of the root query holds a key in understanding the user's intent while searching. There is a lot of research done on capturing query reformulations for query expansion. Einat Amitay et al [21] describe the success of their approach in expanding the index using query reformulations in a session to serve further queries.

We look for capturing query reformulations in a federated search setup. In a federated search environment, a user's query context differs for different media types. The following example illustrates this point. A user with intent of *visiting Mauritius* seeks different information about *Mauritius* in different search engines. He searches for *best time to visit Mauritius* in Yahoo Answers. On the other hand, he searches for *cheap hotels in Mauritius* in Yahoo Web Search. Additionally, he searches in Yahoo Image search for *beach pictures of Mauritius*. This example illustrates our point that user's expectation varies from one search engine to another based on their

result type. Another reason for this behavior could be that the user understands what kind of queries work best in which search engine. A user acquires this knowledge based on hit and trial, some intuition and his past search experience.

We illustrate the point of a query trail in federated search by a simple example. Suppose user searches for *php tutorials* in *Yahoo web search*. One of the tutorial results has an announcement of upcoming php tutorial classes at some convention centre. User searches for the *convention centre's address* in Yahoo Maps. Searching for *php tutorials* in Yahoo Maps might not return any relevant result (due to it's crawl frequency, retrieval methods, page ranking schemes, and other ambient characteristics), nevertheless there is a strong link between *php tutorials* and *convention centre address* in Yahoo Maps atleast for this user, if not others. Also, it is not possible to link two different queries fired in two different search engines, as that would require search log analysis which can't be done in real-time. Another point is that it would require user to be logged in, so that his queries can be identified to identify any trail.

In SAFE, we try to capture modifications to a query in federated search environment. To make our task of identification of a reformulation from a root query simpler, we have a main search box from where a user starts. The results of his query are presented in a modular fashion depending upon his selection of modules and our criterion for module placement. A user can modify his query in any of the modules, if the results in that module are not satisfactory but in other modules are. We call this association between the original query and it's modification as *Query Association* and represent it as: query  $\Rightarrow$  modified query (module name).

Query Association gives very useful information about the intent of the query in various modules. The difference between the main query and modified query (which user expects to work better in that module) gives an idea of the context intention of the user. It also gives strong hints on how similar queries can be modified in order to get better results.

#### 3.1 Template Query Association

Query Association is essentially a method to record contextual keywords that can be applied to original query in different modules in order to get more relevant results. There are times when users want similar keywords to apply to different queries which are contextually tied. An example illustrates this point. Suppose a user is planning to choose either of *mauritius* or *seychelles* for his next holiday. If user forms a query of the type *places to visit in mauritius* and *places to visit in seychelles*, then *places to visit in* can be abstracted out in a template, say *bestplaces*, and this template can be applied to both the queries. SAFE supports creation of such association and their application by the user, as discussed in next section.

### 4. SAFE

SAFE (Search Aggregation and Federation Engine) is a meta search engine that uses Yahoo's public api to tie results from different search engine sources such as Yahoo Image search, Yahoo Web search, Yahoo Answers, Flickr, Yahoo Videos, Yahoo Maps, etc. It has a modular user interface, in which every module is tied with unique search engine source and results from different modules are not mixed. Modules are laid out in a grid in the main web page as shown in figure 1.

Every module is associated with a specific search engine source

san francisco search

**Yahoo Search - cheap hotels san francisco** next

CheapTickets.com - Find Cheap Rates on Hotel Kabuki in San Francisco Rates as low as \$129 at Hotel Kabuki in San Francisco View hotel photos, customer reviews, and much more. CheapTickets.com - trusted since 1986

San Francisco, CA Hotels from \$1.99 Low Prices at Orbitz.com Find great deals and exclusive rates on top San Francisco hotels. Sort hotels by price, view hotel photos, amenities and more.

**Yahoo Videos - san francisco parade** prev next

2006 San Francisco Chinese New Year Parade-1V43E  
2006 San Francisco Chinese New Year Parade Author:  
shockings Keywords: San Francisco Parade Chinese  
New Year Added: February 12, 2006 Recently Added  
Videos



**Yahoo Answers - pubs in san francisco** next

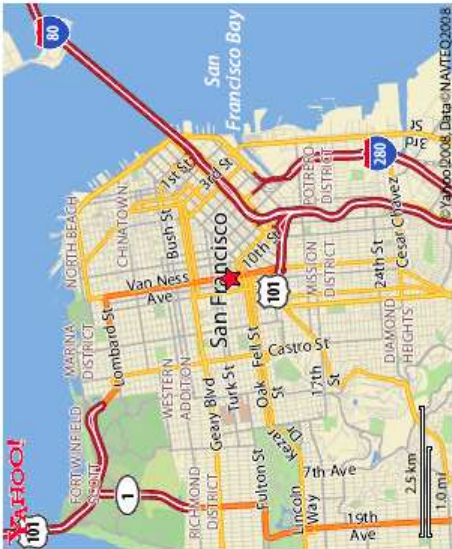
Im looking for a bar or place that will be showing UFC 78 in San Francisco and how much it costs to watch? Im looking for a bar or place that will be showing UFC 78. I got a list of: CA San Francisco Bayside Sports Bar & Grill CA San Francisco Hoollers @ S  
Jillans San Francisco CA 415 369-6100 any of those places are probably free unless they regularly have a cover charge, but it won't cost anymore just because UFC will be showing, its a promotional  
what bars in san francisco have shuffleboard? my friends & I are looking for a fun pub in San Francisco to have a couple beers & play shuffleboard after a round of golf. Any ideas?  
jacks corner in the mission on 24th and Utah

**Yahoo News - san francisco** next

MLB: Cincinnati 10, San Francisco 9  
Brandon Phillips had three RBI Saturday night to lead the Cincinnati Reds past San Francisco 10-9.

san francisco / Workshop, auction illustrate 2 sides of mortgage crisis  
Virginia Quintero and her 59-year-old mother came to San Francisco's Bayview neighborhood Saturday, hoping to save their Pacifica home from the clutches of the bank. Three miles away, Jim Morris of Se

**Yahoo Maps - san francisco** next



**Flickr - golden gate bridge** next



**Image Search - beaches san francisco** next



Figure 1: SAFE presents different modules representing different search engine resources in a grid based layout.



**Figure 2: Module options.**

and it manages query processing and retrieval of results from that search engine using Yahoo Developer Network’s public apis. Each module provides certain useful features as shown in figure 2, enabling user to customize the module, such as a) searching in that module, b) customization of number of results to show, c) collapsing and un-collapsing the module results, d) closing the module. These features appear by clicking the module home button (hence require two mouse clicks to activate) located on top left corner of each module as shown in figure 1.

Modules also implement other search related features which require only one mouse click, such as 1) clicking module name to search in that module (pops a text box where user can type query modification), 2) navigation to next page of results, 3) navigation to previous page of results, 4) rating controls, so that user can give explicit rating to the results. Feature 2, 3, 4 can be seen in Figure 1 (see Yahoo Videos module). Feature 1 enables user to search exclusively in that module. There are several benefits of this feature as mentioned below:

- It doesn’t update the results in other modules. This is useful for comparative analysis and also when user search has a trail and he doesn’t want to lose relevant results in other modules.
- Searching in one module is faster than searching in main search box which leads to searching in all the modules and hence a bit slower. This is due to higher http payload, concurrency issues in firing queries to different search engine in parallel, difference in search engine response times and loading of results on client side (images have to be loaded).
- Normally, the main page doesn’t fit completely in user’s screen, so he has to scroll down to see a module at the bottom of the page. It is faster for user to search in the module he’s focused on than to scroll up and locate the main search box for searching.
- It doesn’t break user’s focus since nothing else changes for him except for the results in that module. The user is as focused on that module as before firing the request.

When a user uses the module’s search feature, SAFE creates a query association between the main query (also referred as root query) and the query fired in that module. As an example from figure 1, user searches for *san francisco* in the main search box and later searches for *golden gate bridge* in Flickr module, so following query association is recorded by SAFE.

san francisco  $\Rightarrow$  golden gate bridge, (flickr module)

SAFE also captures ambient data for a query association. In general, a query association is described as in Eq. 4.1

$$q \Rightarrow \bar{q}, (m, u, t, \epsilon) \quad (4.1)$$

where,

$q$  = root query

$\bar{q}$  = modified query

$m$  = module name

$u$  = user name

$t$  = creation timestamp

$\epsilon$  = rating of association based on results (default value is 1, clicking *thumbs up* image increase rating to 1.25 and clicking *thumbs down* image decreases it to 0.75, as in figure 1.

Apart from creating associations, users can perform following useful activity which helps SAFE in determining rank of an Association:

- Clicking a result.
- Clicking Next to get next set of results.
- Clicking Prev to get previous set of results.
- Giving explicit rating through rating controls (*thumbs up* or *thumbs down* image).
- Changing number of results to show for a module.

User’s activities are recorded along with their timestamp, association on which activities are made and session in which they are performed. When a user logs-in to SAFE a unique session-id is generated. This helps in distinguishing user’s activity in one session from another. For users who do not wish to login, all their activity are recorded as a *default user*. Session for default user expires after one hour.

SAFE provides three searching modes, namely 1) Collaborative 2) Personalized 3) No Association. These modes affect the way in which associations are applied to a user’s root query in the modules. In *collaborative mode*, every module searches for highest ranked query association for the given root query across the system. It considers associations created by all the users of the system in that module and the best possible query association becomes the modified query and module fetches results corresponding to the modified query. In *personalized mode*, only query associations created by this user are taken into consideration to recommend a modified query. In *no association mode*, module fetches results corresponding to the root query only. Query association rank computation is discussed in detail in section 5.

SAFE also allows users to apply a query association as template to other queries. Set of keywords can be triggered as template by wrapping them in *colon (:)*. For Example, Suppose a user has association as shown in table 1 and user form a root query *mauritius :travel:* then his queries are expanded in images and answers module as shown in table 2.

Association	Module
travel $\Rightarrow$ beach pictures	images
travel $\Rightarrow$ cheap hotels	answers

**Table 1: Associations for the query term *travel***

Association	Module
mauritiu :travel: ⇒ mauritiu beach pictures	images
mauritiu :travel: ⇒ mauritiu cheap hotels	answers

**Table 2:** *travel* used as template for query term *mauritiu*

The main advantage of searching using template associations is that it can make searching faster for users. If a user is searching for various related terms and every time he is performing same query modification to each of them (to get better results) then creating one association and using it as template can save considerable time and effort. Template associations are strong indicator of what query modifications can be applied to different queries belonging to same category or context. SAFE uses that to recommend query modification when user searches with either collaborative mode or personalized mode on (details in section 5).

SAFE also implements a module ranking scheme. In this, all the modules are ranked for a given root query based on its query association and query associations of queries belonging to the same category as this root query. Based on the individual rank of a module it is laid out in the graphical user interface. Module with higher rank is placed above or before the module with lower rank. Section 6 talks in detail about the module ranking scheme.

## 5. QUERY ASSOCIATION RANKING

SAFE ranks query associations in order to recommend best possible modification for a root query in all the modules. Since the number of associations can be large and ranking process is time consuming, SAFE delegates rank computation to a background process. The background process updates the rank periodically after short time intervals, in order to ensure freshness of recommended modifications by taking into account newly formed associations and user activities.

Query Association ranking is based on LRFU Block replacement Policy [23], which combines recency and frequency to quantify the likelihood of a block to be referenced in future. Query Association ranking is computed based on user’s activity on a given association. When a user performs an activity (such as create association, click result, click next, click prev, give explicit feedback, change number of results) on an association, we consider that as an reference to the Query Association, analogous to reference to a block as described in [23]. Each reference contributes to the likelihood of the association and it’s contribution is determined by the *weighing function*  $F(t)$  where  $t$  is the time span from the reference in past to the current time. More formally,  $F(t)$  is given by equation 5.1.

$$F(t_{now} - t) = 2^{-\lfloor \frac{t_{now} - t}{t_{now}} \rfloor} \quad (5.1)$$

where  $t$  is the timestamp of activity,  $t_{now}$  is the current time. Query Association activities also record user’s session-id. This helps in distinguish activities made by user in one session from another session. For Query Association rank computation, activities in only last three sessions in which a query association is referenced are considered. For example, a particular Query Association is referenced in user session 1, 5, 8, 11, 21. Then, activities made in session 8, 11, 21 are considered for ranking this association. This policy has several advantages such as it reduces the number of activities that need to be stored for an association. It also helps query

association to recover from a low initial rank (since rank is based on reference time). It also ensures that our ranking is resistant to profile hacks and malicious user activity (because eventually malicious activity might get rolled out after user’s profile is restored and user starts searching normally).

Following algorithm describes how rank for a single query association is calculated.

---

### Algorithm 1 Query Association rank calculation algorithm

---

Lets assume that for a root query  $q$ , there are  $N$  unique associations  $q_1, q_2, \dots, q_N$  for a given module  $m$ . Unique associations implies  $\{q_k \neq q_l\} \forall k, l \in N, k \neq l$

Let  $A_i^m$  represent association  $q \Rightarrow q_i$  in module  $m$ . Lets assume that  $A_i^m$  is referenced by  $M$  users  $u_1, u_2, \dots, u_M$ . Let  $R_{ij}^m$  represent rank of association  $A_i^m$  as per user  $u_j$ .  $R_{ij}^m$  is given by following formula:

$$R_{ij}^m = \epsilon_{ij} \times \frac{1}{K} \times \sum_{k=1}^K F(t_{now} - t_j^k) \quad (5.2)$$

where,

$\epsilon_{ij}$  is explicit rating of *association*  $A_i^m$  given by *user*  $u_j$  (default value is picked if user hasn’t given explicit rating).

$k$  represents  $k^{th}$  activity of *user*  $u_j$  that references *association*  $A_i^m$  (Also,  $k^{th}$  activity belongs to one of the latest three sessions in which *association*  $A_i^m$  is referenced).

$t_j^k$  represents timestamp of *activity*  $k$  of *user*  $u_j$ .

$t_{now}$  is the current time.

Note that if a user is not logged in then all his associations belong to a default user and session for default user expires after one hour. So all Query associations made by anonymous users in that hour belongs to the same session.

Note that for a logged-in user, there must be some activity on  $A_i^m$ . If there is no activity then that means that user hasn’t referenced  $A_i^m$  and this user is not part of  $M$  users who referenced  $A_i^m$ . It is possible that SAFE has recommended  $A_i^m$  to this user but user choose to overlook it.

$R_i^m$  represent rank of association  $A_i^m$  as per all the users of the system.  $R_i^m$  is the sum total of rank of all the users who referenced *association*  $A_i^m$ .  $R_i^m$  is given by the following formula:

$$R_i^m = \sum_{j=1}^M R_{ij}^m \quad (5.3)$$

If a user  $u_j$  has fired the root query  $q$ , then  $A_i^m$  is ranked  $R_{ij}^m + R_i^m$ . SAFE picks the highest ranked association and uses that to retrieve results in module  $m$ .

If user’s searching mode is set to personalized, then  $A_i^m$  is ranked  $R_{ij}^m \times k$  for this user, where  $k$  is number of association activities made by user  $j$  for association  $A_i^m$ .

---

Additionally, SAFE keeps track of queries used as template keywords (queries wrapped in *colon*). Root queries on which template keywords are applied (root query is obtained by removing all tem-

plate keywords from the original query) are categorized based on directory categories recommended by *open directory project* (also known as *dmoz*) [8, 9]. Templates get associated with the *dmoz* categories of the root query. For a given category, list of templates that are associated with it is maintained along with all the reference timestamp of the template. Rank calculation for template follows the same formula as described in equation 5.2 with a difference that instead of activity time we have template reference time and there is no explicit rating of a template.

For a root query with no query associations, it's most ranked *dmoz* category is fetched (there can be many *dmoz* categories for a single query, we pick the one with maximum match). Highest ranked template for this category is applied to the root query and results are fetched accordingly in all the modules. If *personalized mode* is on then only templates referenced by this user are taken into consideration and rank is computed based on references created by this user only (same as described in algorithm 1).

## 6. MODULE RANKING

Modules are ranked for root queries based on association activities and user preference. SAFE arranges modules in its graphical user interface based on module rank. Module ranking takes into account following factors:

- **User Preference:** If a user has collapsed a module, it is rearranged at the bottom of the page. If a user has closed the module, then it doesn't appear in the user interface. Irrespective of module rank, user preference is given utmost priority.
- **User's Query Association:** Query Association created by user in a module indicates user's interest in that module for that query.
- **Community's Query Association:** Query Associations created by community of users in a given module indicate their interest in that module.

Every user activity is associated with a query association which in turn is associated with some module. So for a given query association there is an associated user, module, list of user activities. We prune user activities based on session-id and keep activities belonging to only three sessions (latest three from the list). Additionally, we use *dmoz* category to classify a root query (as described in later part of section 5). More formally, a query association is associated with a module, set  $\theta$  and set  $\Psi$  as shown below.

$$q \Rightarrow \bar{q}, (m, \theta, \Psi) \quad (6.1)$$

where,

$q$  = root query

$\bar{q}$  = modified query

$m$  = module name

$\theta = \{(u, \Lambda_u) : u \text{ is the user and } \Lambda_u \text{ is the list of activities performed by this user on given association in module } m\}$

$\Psi = \{\text{list of categories of root query } q\}$

$\theta$  is also called as activity set.

Module  $m$  is ranked for a given *dmoz* category  $\psi$ , as given in following equation:

$$m(\psi) = \sum_{a \in A} \frac{\sum_{(u, \Lambda_u) \in \theta_a} \frac{1}{|\Lambda_u|} \sum_{\lambda \in \Lambda_u} F(t_{now} - t_u^\lambda)}{|\theta_a|} \quad (6.2)$$

where,

$A$  = set of association as given in equation 6.1 which belong to module  $m$  and *dmoz* category  $\psi$

$\theta_a$  represents activity set of association  $a$

$F$  is the weighing function given by equation 5.1

$t_u^\lambda$  represents timestamp of activity  $\lambda$  by user  $u$

When a user types in a new root query, it's most ranked *dmoz* category is fetched (one with maximum match). Modules are arranged in a sorted order in a list based on their ranks in that particular category and the list is returned back to SAFE so that it can rearrange the modules. If *personalized mode* is on then only activities of this user are taken into consideration. Since module ranking is a time consuming process, a background process computes module rank for all categories by all the users (It also pre-computes the personalized module rank for all the users).

## 7. RESULT AND EVALUATION

We evaluate our meta search engine on the following concepts:

- **Query Association usefulness:** We measure how useful a query association is for the whole community of users searching for the similar concept.
- **Module Placement:** We also measure how good our module prioritization and ordering is.

In order to evaluate the claims of our approach we formed 2 sets of users, 1) Experts, 2) Non experts. Experts have full prior knowledge of the system and knew the benefits of forming associations. Non experts did not have prior knowledge of the system and were asked to evaluate associations by querying the system. We selected 12 experts and gave them in-depth training of all the features and working of SAFE. We got 25 non experts who volunteered to rate the system.

We selected four broad level categories (as mentioned in table 3) and picked 3-4 topics in each of them. We formed 8-10 questions for each of the topics. The questions were picked up from several sources on the web and predominantly from TREC [12], Yahoo Answers, *Are you smarter than a 5th grader?* (a popular TV show).

No.	Category
1	Personalities
2	Places
3	Event
4	Random

**Table 3: List of categories**

We held a competition in which the experts were asked to answer questions from 12 different topics belonging to the above four categories. We played 6 rounds, in every round the experts were given 2 subtopics picked in a way that a topic is assigned exactly to two experts in a round (ensuring that expert has not been assigned that topic previously). Our rounds were time based and we rated the

experts on the basis of their completion time and retrieved information. Table 4 contains list of questions for the topic *Barack Obama*, belonging to category *Personalities*. Table 5 contains list of questions for the topic *Mauritius*, belonging to category *Places*.

No.	Source	Questions
1	TREC	In what US state was Barack Obama born (locate in map)
2	TREC	On what date was Barack Obama born
3	TREC	What year was Obama elected to the US Senate
4	TREC	Whom did Obama defeat for the US Senate seat
5	TREC	What position did Obama hold before becoming US senator
6	TREC	In which US states has Barack Obama lived (locate in map)
7	TREC	What is the last state that Barack Obama won
8	-	Give a picture of Obama with his opponent
9	-	Get a video of Barack Obama in Google
10	-	Name the books written by Barack Obama

**Table 4: Questions on topic *Barack Obama***

No.	Source	Questions
1	-	How is the weather of Mauritius in May ?
2	Yahoo Answers	Which is better holiday spot Seychelles or Mauritius ?
3	Yahoo Answers	What are the must see places in Mauritius ?
4	-	What is the color of water of best beach in Mauritius
5	-	What is the exchange rate in Mauritius ?
6	Yahoo Answers	Name a few good hotels to stay in Mauritius ?
7	-	Give 3 images of Mauritius beach
8	-	Who is the Prime minister of Mauritius
9	-	Where is the city market in Mauritius (locate in maps)
10	Yahoo Answers	Cheap flight ticket deals to Mauritius

**Table 5: Questions on topic *Mauritius***

In the initial few rounds, experts formed more associations in order to discover the answers for the questions. However towards the end, we observed that the number of associations formed drastically reduced as SAFE automatically recommended associations and retrieved relevant results based on associations created by experts in previous session on the same topic.

Table 6 and 7 mention some of the top associations formed by experts after six rounds for *Barack Obama* and *Mauritius* respectively.

No.	Association	Module	No. of Activity	Score
1	honolulu, hawaii	maps	11	8.22
2	barack obama hillary clinton	images	12	8.13
3	obama us senate defeated	web search	11	8.01
4	last state obama won	answers	10	7.53
5	barack obama google interview	video	9	6.1
6	barack obama victory	news	6	3.2
7	audacity of hope	web search	2	1.2
8	barack obama wiki	web search	1	0.53

**Table 6: Associations formed for topic *Barack Obama***

No.	Association	Module	No. of Activity	Score
1	best mauritius beach	web search	15	10.5
2	prime minister mauritius	web search	14	9.2
3	grand bay beach mauritius	images	11	8.22
4	is mauritius better than seychelles	answers	9	7.26
5	seychelles or mauritius	answers	10	7.21
6	mauritius may weather	weather	1	0.53
7	mauritius exchange rate	finance	1	0.44
8	city market, mauritius	maps	1	0.31

**Table 7: Associations formed for *Mauritius***

We found that for *Barack Obama*, associations converged and there were not many competing associations per module. This establishes the fact that experts agreed on previous experts' associations. Whereas for *Mauritius* competing associations were high. Main reasons was that for many questions experts searched in only one module (*web search* or *answers*), reformulating his queries again and again, hence competing associations were there and activity on them were high on average. For some questions, there were only one association formed and the results had enough elaborate snippet that other experts didn't even have to click the result to read and find answer inside (*weather, finance, maps*). Another idea to reduce competing associations is to pick top 2-3 associations for a module and merge results from them instead of picking the top most association only.

Figure 3 shows comparison of number of association activities done by experts in different rounds. Number of association activities decreased as rounds increased. This establishes the fact that users found previous association useful and did not create associations of their own or performed activities on existing ones.

In the next experiment, we tried to compute the precision of the query associations formed in the first experiment. We asked a group of non-expert users to rate the associations based on the results retrieved for those associations and how good a match are the results in solving the questions. Table 8 contains the precision values of

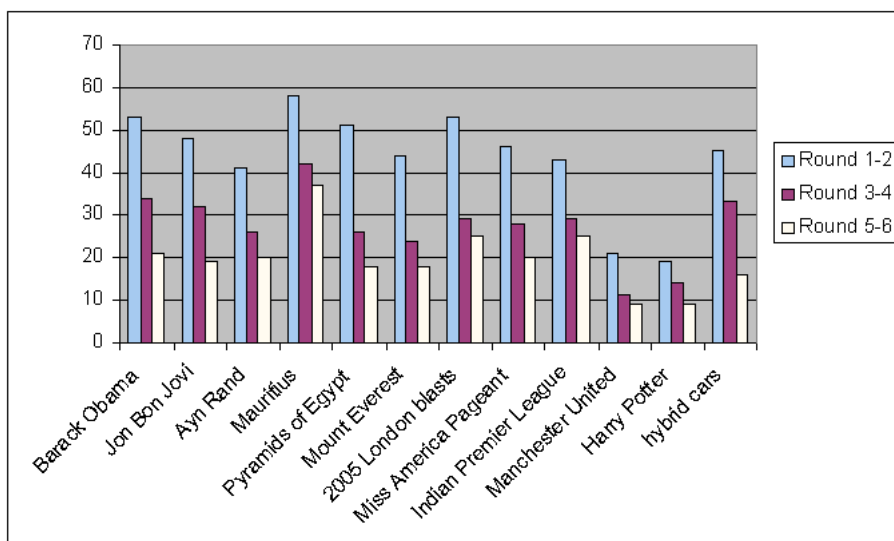


Figure 3: Topics and corresponding association activity in different rounds

all the associations in different modules. We measure precision as the number of relevant associations marked by a user to the total number of associations retrieved for that query in a given module.

Module	Precision
weather	1.0
finance	1.0
maps	0.99
images	0.99
web search	0.95
answers	0.91

Table 8: Precision of Query Associations in different modules

In another experiment, in order to evaluate Module ranking provided by our system, we asked our users to search for the answers to the same queries outside our system, using all the possible available modules. We then examined a query trail to look at the number of times the users used a module and the order in which a module was called. We found out that modules searched by a user were in coherence with the module ranking provided by our system.

## 8. CONCLUSION AND FUTURE WORK

In this paper we built a meta search engine which combines various heterogeneous search engines to a single interface. We propose a scheme to leverage query modification using associations. Our preliminary experiments showed significant improvement in search experience, as per user feedback. With the experiments we could establish the fact that there exist commonalities in queries and their modifications as formed by expert users and we can successfully tap it for a normal user.

We successfully demonstrated the commonalities in queries for the experts and how they can be tapped for the benefit of common user. We would also like to study collaboration between users belonging to communities based on interest, age group, demographics.

We would like to evaluate the benefit of template associations to the user. How it saves user search time. We would also like to

evaluate how template associations are recommended for distinct queries belonging to the same categories.

We would also like to study the effects of a spam user in the system. How we can automatically detect spam activity and how we can cluster users based on some confidence metrics, enabling us to distinguish expert and novice users.

We also wish to reconcile associations which have the same meaning. For example, our system considers these two associations different 1) obama => obama born 2) obama => obama birth. Even though born and birth lead to same set of results. It would be useful to reconcile associations with same meaning as one.

In our present system, we pick the top most association for a module and show the results accordingly. It might be useful to retrieve results for top 3-4 associations and merge results for them and then present in a module. We could also show top k associations in as *also try* link.

## 9. REFERENCES

- [1] Clusty meta-search engine. <http://clusty.com/>.
- [2] Dogpile meta-search engine. <http://www.dogpile.com>.
- [3] Federated search wiki. [http://en.wikipedia.org/wiki/Federated\\_search](http://en.wikipedia.org/wiki/Federated_search).
- [4] Flickr. <http://www.flickr.com>.
- [5] Google's web history. [www.google.com/psearch](http://www.google.com/psearch).
- [6] Mamma meta-search engine. <http://www.mamma.com/>.
- [7] Meta-search: More heads better than one ? [http://news.zdnet.com/2100-9588\\_22-5647280.html](http://news.zdnet.com/2100-9588_22-5647280.html).
- [8] Open directory project. <http://www.dmoz.org/>.
- [9] Open directory project's wikipedia page. [http://en.wikipedia.org/wiki/Open\\_Directory\\_Project](http://en.wikipedia.org/wiki/Open_Directory_Project).
- [10] Search engine watch. <http://blog.searchenginewatch.com/blog/041111084221>.
- [11] Search mash up. <http://www.searchmash.com>.
- [12] Trec. <http://trec.nist.gov/>.
- [13] Yahoo! alpha search. <http://au.alpha.yahoo.com/>.

- [14] Yahoo answers. <http://answers.yahoo.com>.
- [15] Yahoo! developer network. <http://developer.yahoo.com/>.
- [16] Yahoo images. <http://images.yahoo.com>.
- [17] Yahoo maps, [www.maps.yahoo.com](http://www.maps.yahoo.com). <http://maps.yahoo.com>.
- [18] Yahoo news. <http://news.yahoo.com>.
- [19] Yahoo search. <http://search.yahoo.com>.
- [20] Yahoo videos. <http://video.yahoo.com>.
- [21] AMITAY, E., DARLOW, A., KONOPNICKI, D., AND WEISS, U. Queries as anchors: selection by association. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia* (New York, NY, USA, 2005), ACM, pp. 193–201.
- [22] III, H. D. Web search intent induction via automatic query reformulation.
- [23] LEE, D., CHOI, J., KIM, J. H., NOH, S. H., MIN, S. L., CHO, Y., AND KIM, C. S. Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies. *IEEE Trans. Comput.* 50, 12 (2001), 1352–1361.
- [24] LI, L., AND SHANG, Y. A new method for automatic performance comparison of search engines. *World Wide Web* 3, 4 (2000), 241–247.
- [25] MA, Z., PANT, G., AND SHENG, O. R. L. Interest-based personalized search. *ACM Trans. Inf. Syst.* 25, 1 (2007), 5.
- [26] PITKOW, J., SCHÜTZE, H., CASS, T., COOLEY, R., TURNBULL, D., EDMONDS, A., ADAR, E., AND BREUEL, T. Personalized search. *Commun. ACM* 45, 9 (2002), 50–55.
- [27] QIU, F., AND CHO, J. Automatic identification of user interest for personalized search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web* (New York, NY, USA, 2006), ACM, pp. 727–736.
- [28] RIEH, S. Y., AND XIE, H. Analysis of multiple query reformulations on the web: the interactive information retrieval context. *Inf. Process. Manage.* 42, 3 (2006), 751–768.
- [29] SHANG, Y., AND LI, L. Precision evaluation of search engines. *World Wide Web* 5, 2 (2002), 159–173.
- [30] SI, L., AND CALLAN, J. A semisupervised learning method to merge search engine results. *ACM Trans. Inf. Syst.* 21, 4 (2003), 457–491.
- [31] SI, L., AND CALLAN, J. Modeling search engine effectiveness for federated search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM, pp. 83–90.