10.31341/jios.45.1.13
Open Access

UDC 004.738.52
Original Scientific Paper

Real-time Web Search Framework for Performing Efficient Retrieval of Data

Falah Al-akashi *falahh.alakaishi@uokufa.edu.iq*
Faculty of Engineering
University of Kufa, Najaf, Iraq

Diana Inkpen *Diana.Inkpen@uottawa.ca*
Faculty of Engineering
University of Ottawa, Ottawa, Canada

Abstract

With the rapidly growing amount of information on the internet, real-time system is one of the key strategies to cope with the information overload and to help users in finding highly relevant information. Real-time events and domain-specific information are important knowledge base references on the Web that frequently accessed by millions of users. Since real-time data have only a short time to live, real-time models have to be continuously adapted, ensuring that real-time data are always up-to-date. The focal point of this manuscript is for designing a real-time web search approach that aggregates several web search algorithms at query time to tune search results for relevancy. The evaluation showed that the proposed approach outperforms the traditional models and allows us to adapt the specific properties of the considered real-time resources. Compared with offline approach, Wikipedia implication was highly improved the relevancy of our real-time approach, 0.405 for precision and 0.430 for nDCG, the average search duration was 4.9 ± 3.2s (N = 1000 searches or queries). The mean search duration to each individual resource was between 0.05s and 4.55s. The average system runtime or system overhead was 0.12s, whereas, the deadline for receiving responses from all resources over the network was 10s. Overall, the proposed outcomes were significantly better than those available in similar methods presented previously, it is highly relevant for today searched queries, consistent in its performance, and resilient to the drawbacks faced by other algorithms.

Keywords: Wikipedia, Resources Correlation, Federated Search, Web Mining, Vector Space Model

1. Introduction

Crawling and indexing immense volume of web content has been a new challenge. Web has drastically moved the way of presenting and anticipating information. Its international ecosystem of applications and services platform allows us to search, aggregate, combine, transform, replicate, cache, and archive the information that underpins today’s digital society. Despite its chaotic growth and having barely entered its serious years, it's the biggest and least formal integral project ever attempted. Today, users are the direct consumers of the services offered by the majority of today’s web applications. Given its success in managing user information needs at such phenomenal scale, this raises the question: how the Web’s underlying architectural would be used to building a federated system that able to deal with such kind of massive data, particularly the kinds of distributed systems typically implemented by enterprise applications and developers? One solution to deal with such phenomena is to use a federated index platform which is a special kind of array that ensembles similar topics of documents in each vertical, such index is generic in nature that efficiently lookup any of the index-key topics [1]. To involve this technique, web search algorithms use a web crawler which is an internet bot that systematically browses content and collects data. Due the volume of internet is growing up, the largest crawler might unable to create a complete index and search algorithms have extreme difficulties in finding relevant search results. Although this is slightly improved by modern search models and relevant results are now available instantly, they are still not refreshed adequately. To address this, some search models have used different crawlers for different topics; for example, an academic-focused crawler is used to access only academic-related documents. Although the process of indexing prioritizes of one task over another, indexing the whole collection created in different periods is highly challenged. This means that the collection is still not refreshed unless the indexing process is created at query time. As a consequence, searching the internet with broad queries tends to produce results or conclusions that differ systematically from the truth [2]. Such search technique is knotted as the quality of data in each web resource that cannot be assessed through analyzing its content alone [3].

In this paper, we tend to use a canonical real-time model for collaborating information on the basis of different information computations between web resources. Studying the impact of information quality value and communication cost aims to show different resources are optimal for different informational need. Within the context of real-time system design, we tend to be able to address many analysis problems like handling words with multiple meanings (polysemy), user modeling, cooperation among agents, and communication between user and resources. It’s necessary to grasp the state-of-the-art applied to information quality and information optimization in Web resources to conduct a study regarding the benefits and shortcomings that the system presents. Different resources used in every topic means different algorithms are involved since each resource associates with different features for its algorithm. Despite researchers have recommended using authoritative indexes for specific queries, but indices still require more information for each resource to produce reliable content. Learning to rank tends to be a solution to narrow diverse queries but it could not distinguish between authoritative and non-authoritative content [4]. As a result, the probabilistic models for task specific category based grasping had been alternative solutions. Web connectivity is relevant topically and search services have turned to information of resources rather than web content itself, Ding et al. [5] and Lawrence and Giles [6] and Kaptein et al. [7]

showed how the external resources were important to improve the quality of data. However, the federated search is the last attempt to handle such problem, but generalizing Web search task involves more challenges than federated search for specializing Web search task [8]. Various resources in common topics mean encompassed several different algorithms because each resource is usually treated with combination of different features. Grouping high-level resources in different topical federated verticals is very important at this time. This helps efficient adhoc queries without need to create different indices for different sets of topics. Indexing real-time data for real-time query searches is challenge since new data, e.g. micro-blogs, news, tweets... etc. created in a short term meanwhile user query functions remain permanently changeable over time [9]. However, we aim to address all discussed challenges by exploiting the adaptation technology for developing our previous offline web search algorithm, which was the best ranking model according to the TREC evaluation campaign, to real-time search model.

The rest of paper is organised as follows: Section 2 outlines some related works to our model. Section 3 describes how we formulated our approach by discussing features and other signals exploited in the algorithm. Section 4 details the experimental results. Section 5 lists some challenges that faced our experiments, and finally, Section 6 will provide conclusion remarks

2. Related Work

Federated search, social search, real-time search and aggregated search are kinds of adaptive searches. Building an efficient adapted learning model remains challenge and several concerns faced this obstacle such as communication, scalability, heterogeneousness of data, and privacy. Researchers addressed some of the communication and scalability challenges in federated learning using some efficient communication in federated learning methods with provable performance guarantees. Meta adaptive index introduces valid alternatives for a large number of non-adaptive indexes or specialized indexes and improves runtime, robustness, and convergence speed over the standard methods [10]. Adaptive indexing in real time search e.g. twitter was proposed by [11] to provide trends including recommendations, recognition, and manual and automated searches. Arya et al. [8] proposed a personalized federated search at LinkedIn; they proved that federated searches for generalized Web search engines present more problems than federated searches for specialized tasks. Some heterogeneous methods simulate large set of particularized indexes. Experimental tests showed how the traditional real-time indexes were comparable with the traditional offline index, and there was a superior performance with different workloads and an average speed limit 2x. However, several researchers were argued for demonstrating the importance of adaptive search, e.g. [1, 12, 13, 14, 15, 16]. Nevertheless, unlike traditional Web search engines that typically blend either a block of results or individual results from different verticals [17]. Some models used particular offline data and resources for their indexes, researchers [22] explored additional idea for building an efficient index over offline Web page collection while their algorithm ranked as the best run in the Web diversity task. They managed the collection and exploitation of knowledge in Wikipedia for enhancing the relevancy of topics' predictions and findings. Our proposed approach exploited this technique for aggregating individual results from verticals and/or blocks of verticals, normalized some difficult features, and finally, produced relevancy rank which is comparable across verticals' results and its features (i.e., individual vs. block). When we started to scrape results from independent Web search resources in all predictive kinds, we hoped to inspire a research that is able to come up with elegant and efficient solutions to the distributed search. To the best of our knowledge is all existing real-time indexing techniques focus on creating only single dimensional indices that is not suited to an efficient and effective task. Traditional models fail to index large size of data in real-time, or sometimes, fall short of providing flexible data retrieval capabilities and scalable indexing services [1]. The University of Glasgow [27] used reformulations algorithm for the user's initial query with a Terrier data-driven learning algorithm which was a learning-to-rank logistic regression platform for the fast computation of document features. The model was based on gradient-boosted regression trees. In logistic regression, interaction could be included through various degrees of terms' interactions, and consequently, the uogTr algorithm technically faced computational challenges with model estimation and poor fit. The exponentially increasing Internet contents along with the rapid expansion of Web applications caused the problem of information overload. One solution to resolve the above problem is to personalize the network applications for individual users [18]. However, recommending web pages for users is another solution; Kim [19] proposed a system that learns user interests by reading the user's bookmark items and monitoring user's behaviour. Based on the automatically constructed user profile, it collects and filters web pages to recommend related web pages as bookmark items.

3. The Proposed Approach

The goal of our approach, as shown in Figure 1, is to achieve and benefit from the popular global indices available in the public resources over the Internet while the cost of index creation is the main assumption. While hiding or minimizing index is important, it seems to turn side effects to read only queries into update transactions which sometimes form locking argument. Despite all the difficulties and the challenges that faced, concurrency controlling in the context of federated indices is very important. Implementing and controlling federated network rigorously separates index structures from index contents which minimizes the requirements and other constraints during index creation. It confirms the fact that federated indices iterated continuously and take advantages of concurrency assumptions. However, federated index must: (a) preserve its advantages even when evolving to run synchronizing queries, (b) utilize the parallelism opportunity for concurrent queries, and (c) follow the overheads of federated behavior, e.g. the number of concurrent conflicts and other concurrency assumption. The federated search requires centralized coordination of the searchable resources, and this assumption does not exist in distributed search. This involves both coordination of the queries transmitted to the individual search algorithms and fusion of the search results returned by each algorithm. Our proposed algorithm is composited from several resources bunched topically in labeled verticals. The technique for selecting the relevant resources and grouping them in verticals will be discussed below.

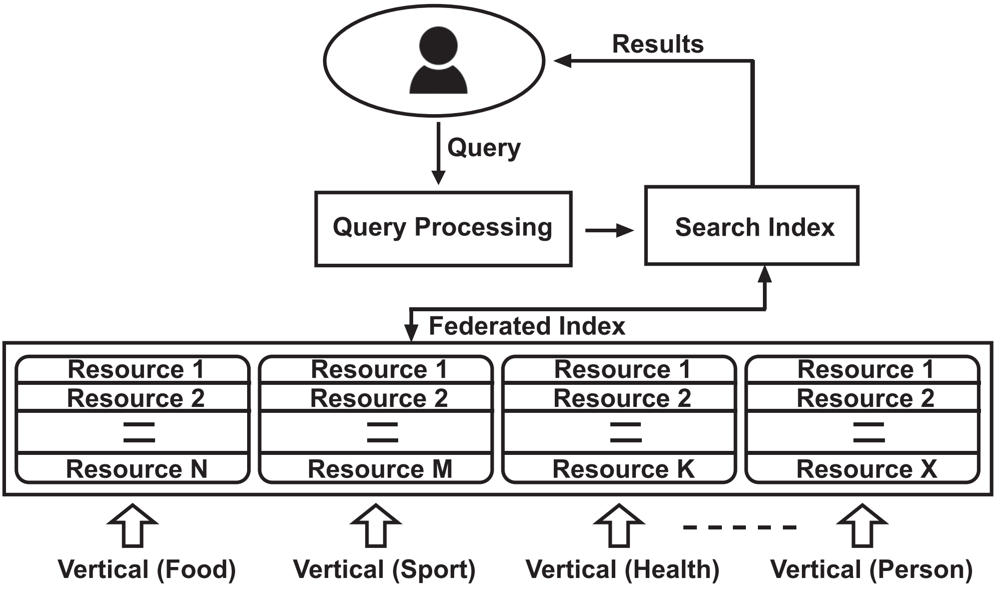


Figure 1. Our overall approach

3.1. Verticals Modeling

Classifying and grouping articles topically into verticals is important to reduce the margining of user’s needs. Labelling vertical topics is an essential step in the federated indices [20]. Determining topics in the distributed resources requires a pool of intelligent data be able to classify the web resources into topical verticals. In Internet, knowledge is controlled by some important resources e.g. Word Press, Wikipedia, etc. Wikipedia is a knowledge-based system that continually being revised and updated, and articles on historic events can be accessed within minutes rather than months or years. Since professionals can update and improve it, Wikipedia has become the most comprehensive knowledge and information resource to date. In addition to quantity, contributors also address content quality, which makes Wikipedia a continuous work-in-progress with millions of articles in various stages of completion. As articles developed, they tend to become more diverse and balanced, and the quality improves over time as misinformation and other errors are removed or repaired. Due to these characteristics, our approach to vertical classifications is based on the relationship between the Wikipedia content and topic specification. Wikipedia has a list of major topic classifications that used to organize the presentation of links to articles in various categories[1]. First of all, all articles in Wikipedia corpus have been read to tokenize the first blocks and extracting the required keywords. For example, the first block in the 'lymphoma' article is "*Lymphoma is a group of* ***blood cancers*** *that develop from lymphocytes (a type of white* ***blood*** *cell). The name often refers to just the cancerous versions rather than all such* ***tumors***". Hence, the target keywords are '*cancer*', *blood*', '*cell*', and '*tumor*', and all keywords are related to a topic "*disease*" which later be label of vertical. Another example, a query "University of California" returns "*The* ***University*** *of California (UC) is a public* ***university*** *system in the U.S. state of California. Under the California Master Plan for Higher* ***Education****...*"; the target keywords are '*Education*' and '*University*'. Table 1 shows sample of vertical names and the relative keywords.

Keywords	Verticals
Actress, maker, player, born, former, president, minister, …	People
Benign, tissue, tumor, disease, diagnosis, cancer, blood, stomach, …	Disease
University, School, Faculty, Institution, Academy, Education, …	Academic
Diet, Overweight, Sport, Olympic, game, activity, skills, racing, …	Sport
Tax, Property, Money, Finance, Fund, taxpayer, income, payment, ...	Finance
Brand, Goods, Hardware, Software, Game, Machine, Money, …	Shopping
Sport, Diet, Overweight, Food, Recipe, Health, Well-being,….	Health
Why, What, How, Where, Cause, Reason, Fact, Question, Answers, …	Q /A
Country, City, Capital, State, Location, Place, Geography, Street,, …	Place

Table 1. Vertical names and its keywords

However, computing the impact of keywords in the first block of each article refers specifically to term impact in the entire document content [2]. Such an algorithm is relevant because Wikipedia also computes the relevancy of terms in the documents to classify their contents into different topics. The algorithm below shows pseudo codes for assigning vertical names.

ALGORITHM 1: Vertical Modeling

```
Inputs:   Individual keywords from Wikipedia (W) and Query (Q)
Output:  Aggregated Resources in Vertical
          Q ← query string, W ← Wikipedia vector, V ← Vertical, S← Resource
          t[ ]← tokenize Q, f[ ] ← tokenize W
```

[1] https://en.wikipedia.org/wiki/Category:Main_topic_articles

```
        forEach (m) in ($T$), forEach (n) read (f)
        if (m==n) then for i=1 to length(V)
        V→ List.Add(Si)
Return: Resources List
```

3.2. Resources Modeling

Federated search often integrates disparate information resources within a single large organization ("enterprise") or for the entire web. Researchers typically classify web resources and type of user queries into three categories: navigational, information, and transactional [21]. Similarly, information belong to one category is specified to a particular area of interest, and resources belong to the same category were not ranked equally. Global rankings based on 'Alexa.net[2]' ranks resources differently and the weight of resource is equal to the number of visits. Our approach for resource selection is adequate to use this assumption, that is, each vertical selects the top resources that utilize better services than others as reported by a global rank. The key abstraction of data in REST may be a resource. Any data provider is a resource, e.g. document or image, temporal service (e.g. "today's weather in Los Angeles"), group of alternative resources, non-virtual object (e.g. in alternative words, associative conception which may be the target of an author's machine-readable text reference that matches among the definition of a resource. A resource may be an abstract mapping to a group of entities, not the entity that corresponds to the mapping at any specific purpose in time. It is vital to select the proper resources and model them at the proper graininess. GitHub API is an example of a fairly elegant API model that used within the right resources. A resource must capture dataflow and dependencies among the functional elements in federated systems, at that point, each vertical might hold few or several resources belong to similar topics. Given a query, a list of resources and related knowledge about resource feedback and network conditions are important to exploit. The resource selection approach produces a ranking list of resources ordered by the query and the impact on the network conditions. Resource selection includes balancing network resource consumption against response quality [9]. We can look at the properties of specific search terms or introduce ancillary query components such as the information sought. The resource specification algorithm creates search plan in two steps, in the first step, resources are ranked with respect to the query and current network conditions, and in the second step, concurrency must be computed as the number of resources simultaneously searched. Resources are ranked by predicting which has the highest probability to return relevant results, which includes hosting a query score in the server with the expected waiting time and the number of results returned in specific interval. The query score is computed by determining the relevancy weight for each resource that corresponds to the query terms, which means, the weights are divided by the total feedback of resources that returned from a vertical.

[2] http://www.alexa.com

For a user's query, the method includes identifying highly-relevant resources that match the query, wherein each online resource is associated with returned results in particular time interval. Obtaining a respective query-specific score for each resource matches the query, identifying one or more relevant resources according to the query specific scores. Thus, the query score $Q_{r,q}$ of resource 'r' and query string 'q' was computed as:

$$Q_{r,q} = \sum_{t\in q} \frac{w_{t,r}}{W_r} \quad (1)$$

where $w_{t,r}$ is the weight of query term *t* in resource r, and W_r is the sum of all weights for resource *r* matches query q.

The scores were normalized by adding the performance information of resource, by which, the five most recent queries for resource r used to compute the average number of hits H_r and average response time T_r. The performance information of resources was combined in the fraction:

$$\mathrm{P_r} = \frac{H_r}{100\, .T_r} \quad (2)$$

The denominator is multiplied by an arbitrary constant of 100, making the performance data and magnitude smaller than the score data. The goal was to rank the better resources utilizing index data alone, and then normalized with respect to the global world rank using resource popularity rank Pop_r which was utilized at Alexa.net[3]. The two computations were combined to determine the overall rank $R_{r,q}$:

$$R_{r,q}=(Q_{r,q}+P_r)\ /Pop_r \quad (3)$$

Appendix (A) shows the topical classification of our selected resources in each vertical whereas Appendix (B) shows the topical classification of resources in each vertical involved by related approaches.

On another side, simultaneous between resources is the first and foremost challenge that every federated network has to face. The purpose of simultaneous calculation is to reduce the resources costs generated by queries in a period of high network disturbance and machine traffic. Concurrency is inversely proportional to query cost; the higher cost of submitting resource search queries, the fewer search services will be queried. Concurrency is computed from three cost variables: global network load, local CPU load, and query discrimination value. Ideally, long queries mean high discrimination and low CPU cost, and vice versa. In addition, the expected network load should be estimated by timely inter-networks traffic, and since this is difficult to assess, the typical load must be computed at a particular time of day, as shown in Figure 2.

A high value of concurrency contribution occurs during periods of low network load, (e.g. 3:00 am), and vice versa. The discrimination value addresses how many resources are likely needed for each vertical to find a satisfactory response, by

[3] http://www.alexa.net

measuring how specified or generalized the query is against the resources. If a term has a relatively high number of results and contributions of resources (e.g. London), it presumably represents an area with many resources covering the topic; thus, we expect fewer resources will need to be queried. If a query has little or no results, (e.g. Cheilitis), we need to search more resources to find relevant results.

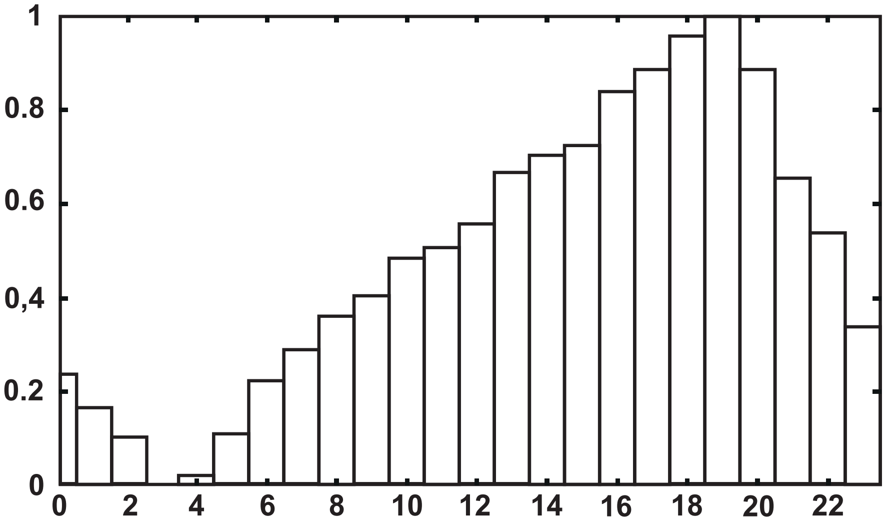


Figure 2. Overall traffic throughout the day

3.3. Query Processing

Query processing is an essential part in the federated search, as it includes: determining the type, searching, and normalizing of a query. To ascertain the type, it was submitted subsequently to three web resources: Wikipedia, Twitter, and Word Press, the responses were analyzed to find the requested verticals. More contrast, submitting a query “lipoma” to the Wikipedia web service using a request REST API; it will return “*A **lipoma** is a benign tumor made of fat tissue. They are generally soft to the touch, movable, and painless. They usually occur just under the skin, but occasionally may be deeper*.” The target keywords are ‘*benign*’, ‘*tumor*’, ‘*tissue*’, and ‘*skin*’ that point to the vertical “*Disease*”. We used a syntactic similarity keywords matching as shown in Table 1 for corresponding vertical name and its topic. If a vector is specified, the system will forward the query to all resources belong to that vertical, wherein the returned results will be stored for strictly ranking. Although term frequency is important for measuring the occurrences of terms and then specifying the topics of documents in resources, it is important to look for different types of document configurations. Two document vectors with the same term occurrences might have different relevancy, because the user preferences are not the same as our current preferences. Al-akashi and Inkpen [22] addressed this by applying ‘term impact’ rather than term frequency. Term impact computes the weight of each term in a vector based on certain features. Each vector was mapped to the resource impact factor computed by cosine similarity between two vectors. The first vector represents the occurrences of terms in a document, and the second represents the query terms. The length of a document vector is inversely proportional to the terms in the query vector. The normalized cosine similarity formula is shown below:

$$\textit{Similarity (D, Q)} = \sum_{j=1}^{m} \frac{(Q_j)^2}{\sqrt{(Q_j)^2} * \sqrt{\sum_{i=1}^{n} tf(Q_i)^2}} \tag{4}$$

where *m* denotes the length of the query terms, *Q* represents the occurrence of terms in document vector *D*, Q_j denotes the occurrence of the query term *j* in *D*, *n* represents the number of terms in *D* and $\mathrm{tf}(\mathrm{Q_i})$ denotes the impact of the query term *i* in *D*.

In order to filter out non-relevant vectors and establish the best similarity impact, our model assigned a strict threshold value that used several sampling vectors to determine only the vectors that had a high impact for all query terms. Below shows a complete algorithm used to resort the aggregated document vectors and generate the final ranking list:

ALGORITHM 2: Result Ranking

Q ← query string, V ← document vector, K ← Scoring, T ← Scoring threshold

- If length (Q) = length (URL) → V = 1.0.
- If length (Q) = length (title) → V = 0.9.
- If length (Q) ∩ length (description) and Q in consecutive location → V = 0.8.
- If length (Q) = length (title) and Q distributed over title → V = 0.7.
- If length (Q) ∩ title; K>T and T>0.5 → V = 0.6.
- If length (Q) ∩ snippet; K>T and T>0.5 → V = 0.5

However, the average query search duration was 4.9 ± 3.2s (N = 1000 searches or queries). The mean search duration to each individual resource was between 0.05s and 4.55s. The average system runtime or system overhead was 0.12s, whereas, the deadline for receiving responses from all resources over the network was 10s.

3.4. Query Expansion

Search engines use query expansion and optimization techniques to enhance the quality of search results and improve efficiency [31]. It is assumed that users do not always formulate search queries to expand the initial results and narrow the final results to satisfy user needs [23]. As a consequence, rather than using fuzzy-wuzzy techniques, external resources play a significant roles in this paradigm, e.g. Word Net, used word sense disambiguation for automatic query expansion for long queries in addition to short queries [11]. Since Wikipedia feedback is suitable for query expansion, researchers [2, 24] exploited the behavior of some Wikipedia writers who adopted the preferences in the dynamic properties of the Wikipedia collection. However, we proposed two techniques, firstly, we used the inter-links between articles, by which, we generally assume that an article 'A' and 'B' are related to each other and shared similar topics if they are inter-linked together. Thus, if a user's query is 'A', the alternative query is 'B' and vice versa. Secondly, we exploited the synonyms and morphologic variation for titling the articles if involved similar contents but titled differently. If the first title matches the user's query and returns few results, the second title will be used alternatively.

However, query expansion and reformulation was not used for all queries, it was used only when the initial ranking list was short and the resulting list contained at least one Wikipedia label that matched the user's query. For example, while the initial results for a query 'angular chelitis' were considered few; the alternative query 'angular stomatitis' would improve the final results. Another obstacle faced our query processing algorithm was fuzzy queries, e.g. 'fibromyalgia', which are one of the most powerful features. To address and resolve this complexity, we applied a computational linguistic algorithm using n-gram string similarity. The n-grams dictionary terms were typically collected from the Wikipedia corpus. For single-word queries we used a unigram model, for multi-words queries we used bigram and trigram models. This helps to search for similar words across hundreds of documents on the index. Wikipedia implication was highly improved the relevancy of our real-time approach even with or without query expansion, the precision was improved from 0.405 to 0.584 whereas the nDCG was improved from 0.214 to 0.430, because Wikipedia was returned 977 out of 1000 results over all queries.

4. Experimental Evaluation

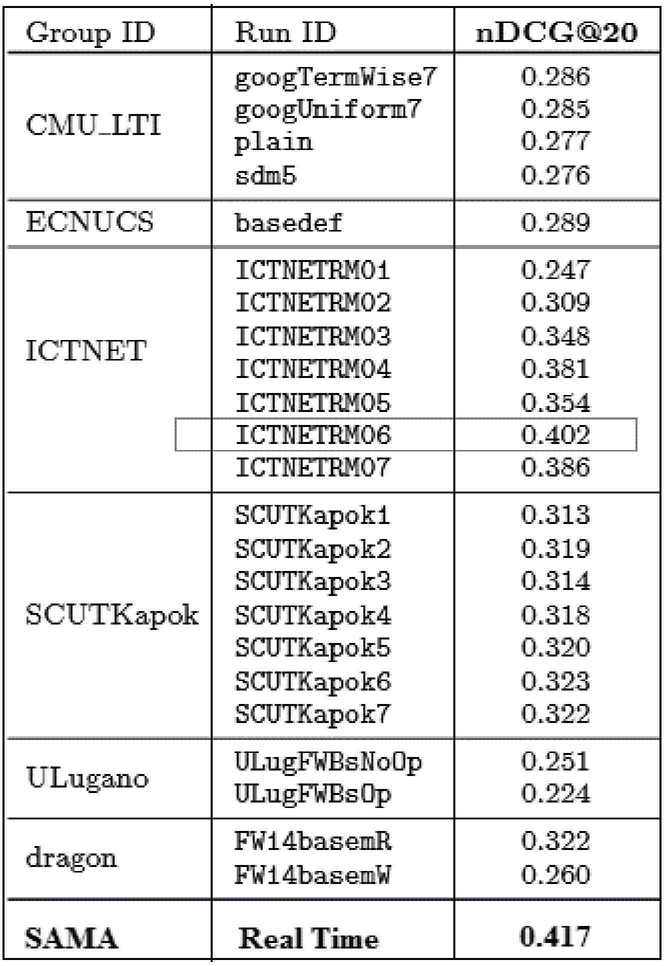
The TREC evaluation campaign provides training and testing sets of queries, whereas the relevance judgments team provides the proposed solutions for each query. Experimentally, web information retrieval approaches evaluated in two tasks: diversity and adhoc. The diversity task is similar to the adhoc task but it differs in the evaluation metrics and judging process. The final goal is to provide a complete coverage and ranked list of pages for a query that aim together to avoid excessive redundancy. The primary effectiveness measure for both tasks is specified by measuring the graded precision of top ten results or graded precision at k, in which documents can be judged as Nav, Key, Hrel, Rel or Non-relevant. The relevancy of selected resource is determined by calculating precision in the subset of results [25, 26]. While the Normalized Discounted Cumulative Gain (nDCG) is the metric of measuring ranking quality for maximizing the relevancy as a whole, it takes into account the graded relevance levels of documents within top ten. Similarly, selecting a best vertical for a given query is determined by running best search query in all resources in that vertical. That means the relevancy of vertical is computed by maximizing the precision of its resources. In the final analysis, the precision of vertical is specified by a threshold since some queries have a small set of relevant verticals, we assumed 0.5 is a threshold of relevant precision). However, if no vertical was selected for a given query, the top vertical with maximum relevancy had been selected as relevant. In terms of testing queries, we used the same set of queries proposed by TREC web track, whereas for training, we used the set of queries in the Million Query track[4]. The testing and training queries involve both tasks adhoc and diversity to represent different complexities of relevancy. Follows is the comparison between our real time algorithm with offline and real time related approaches.

4.1. Comparison with Real-time Approaches

In the information retrieval, comparisons between models must be within similar cluster of training and test dataset. Despite the proposed systems used in our comparisons are little bit old, but it is important to show how real-time search is more efficient than offline search when used with similar dataset and resources. While the similar models used 200 resources distributed among 24 agents, our approach used 59 resources distributed in 20 agents. Recent studies have used single resources for particular topics which are not bearing on the matter being considered except some specifications, and recently, various models have been used large realistic data collections sampled from multitude of online search engines. Table 2 shows the comparison between our mode, Sama, which used custom resources and agents with other top models that used a large collection of agents and resources. The best proposed effective models used similar techniques that aggregate indices from tremendous retrieval algorithms. The best effective models [29] proposed indexes with only documents (rather than snippets) and mix results were improved from the traditional retrieval algorithms like variations; but however, they failed to use external resources efficiently, e.g. Wordnet and Wikipedia. Wikipedia results were showed the content is necessary for features extraction. We have a tendency to assume that for a given query all search results were readily available but more realistic strategies would be initially make selection of a small number of relevant engines to increase recalls and re-rank results again. A notable exception was the RS_clue baseline that used the assortments' of snippets together with the ClueWeb'09 collection to form size estimations. The most efficient model was proposed by the Chinese Academy of Sciences (ICTNET) [30] ranked 0.402 on nDCG@20 scores in all queries. The organizers' baseline runs used the static rankings from the equivalents size-based resource baselines. In our perspective, querying the TREC FedWeb 2013 indices is totally different from the realistic collections. Despite the results of the best run were not qualified well compared to our results, querying several resources at the same time and selecting only five resources deemed the internet to shut down from overload which was not applicable when real-time tasks were involved. Also, one more thing that makes real-time algorithms better than offline algorithms is that in real-time algorithm all spam

[4] The goal of this track is to run a retrieval task similar to standard ad-hoc retrieval, but to evaluate large numbers of queries incompletely, rather than a small number more completely. Participants run 10,000 queries and random of 1,000 or so were evaluated.

documents did not have implication and did not affect the overall retrieval performance because most of documents in the web search resources were filtered out by their administrators. Table 2 shows the comparison between the result of proposed approach "SAMA " and the results of the best approaches that used real-time contents of 157 online resources, including traditional search engines, in order for representing the whole picture of the retrieval methodologies as well as involving heterogeneous content sorts for each approach. The online test set queries is shown on Appendix (C); whereas Million Queries Track[5] was used for training our approach.



Group ID	Run ID	nDCG@20
CMU_LTI	googTermWise7	0.286
	googUniform7	0.285
	plain	0.277
	sdm5	0.276
ECNUCS	basedef	0.289
ICTNET	ICTNETRM01	0.247
	ICTNETRM02	0.309
	ICTNETRM03	0.348
	ICTNETRM04	0.381
	ICTNETRM05	0.354
	ICTNETRM06	0.402
	ICTNETRM07	0.386
SCUTKapok	SCUTKapok1	0.313
	SCUTKapok2	0.319
	SCUTKapok3	0.314
	SCUTKapok4	0.318
	SCUTKapok5	0.320
	SCUTKapok6	0.323
	SCUTKapok7	0.322
ULugano	ULugFWBsNoOp	0.251
	ULugFWBsOp	0.224
dragon	FW14basemR	0.322
	FW14basemW	0.260
SAMA	**Real Time**	**0.417**

Table 2. Overall comparisons between top real time approaches using real time queries

[5] https://trec.nist.gov/data/million.query.html

4.2. Comparison with Offline Approaches

To better focusing on the value and novelty of the proposed approach and its application, we compared our experiment with the best published approaches. For the adhoc task documents were assessed with respect to the topic as a whole. Relevance categories were similar in structure to the categories used in traditional Web search, including spam and junk removal. Also, the top two assessments' structures were closely associated to the home page finding and topic distillation tasks. For the diversity task, documents were assessed based on the subtopics, as well as with respect to the topics as a whole. As a comparison, using assessment metrics shown in Table 3 could achieve the tradeoff between effectiveness (overall gains across queries) and robustness (minimizing the possibility of significant failure, relative to a given baseline). SAMA as our continual improvement of performance that compared with the best models in TREC Web track / the ClueWeb09 collection by using the same set of test queries (Appendix D). The best approach was proposed by [27] at the University of Glasgow (uogTr) used reformulations of the user's initial query with a Terrier data-driven learning model which is a framework for the fast computation of document features. It used with a state-of-the-art learning-to-rank logistic regression algorithm based on gradient-boosted regression trees. Realistically, the predictive model did not capture the complex interactions among the variables in data. In logistic regression, interaction can be included through various degrees of interaction terms, but uogTr approach normally leaded to computational challenges with model estimation and poor fit. Contrary to our proposed method, logistic regression does not provide a way to focus the computations on the smallest subset of variables linking decision variables to the target variable. The second ranked approach "DFalah" used terms impact in document content with query structures [22]. Chinese Academy of Sciences (ICTNET) [28] contacted Learning-To-Rank layer to aggregate several features simultaneously, but the effectiveness was very poor because of the low quality of training data. Quantum Interaction (QI) group proposed information about word associations used first and second order relationships in natural language known as syntagmatic and paradigmatic associations.

Model	**P@20**	**NDCG@20**
University of Glasgow (uogTr)	0.453	0.238
IBM Lab (Srchvrs)	0.315	0.176
QUT's Quantum Interaction (QI) group (QUT_Para)	0.305	0.167
University of Twente (Utwente)	0.221	0.113
Chinese Academy of Sciences (ICTNET)	0.257	0.110
Mu˜gla University IRRA (IR-Ra) group	0.367	0.143
Univerity of Ottawa (DFalah)	**0.405**	**0.214**
University of Ottawa (SAMA)	**0.584**	**0.430**

Table 3. Overall comparisons between the offline runs and our real-time run

5. The Faced Challenges

We hereby introduce four of the problems faced our work with a solution while most of them are related to the communication between resources in the adapted network. These problems make an adaptive- learning realized from other traditional-learning, e.g. adaptive learning available in remotely data in each resource.

- **System Communication:** Communication between a client and remotely resources is a critical bottleneck in adaptive networks since that incorporate with privacy issues over sending raw data. It requires data presented in each resource to be transferred to the user or client and combined concurrently with other results from other resources. Indeed, adaptive data are potentially consisted of a large number of resources, and if tons of resources shared similar topics, it makes communication in the network very slow. In order to make our approach to fit data produced by the resources in the adapted network, it is, therefore, urgently to develop efficient communication algorithm that recursively receive results from resources concurrently as fraction of the training process, as opposed to producing the results in the local resources. To assist minimize communication in such network, three issues are assumed: (1) minimizing the total number of communication cycles, (2) minimizing the size of transferred data at each cycle, and (3) synchronizing the data between resources.
- **Hardware Heterogeneity**: The computation, storage, and link abilities between resources in adaptive networks for providing results might differ because of variability in hardware capability; e.g. CPU, memory, etc., or sometimes a resource communicates with other resources locally. Thus, the adaptive system must afford strong hardware.
- **Resource Trustworthiness:** Most resources share similar topic in a vertical. Some resources might also be unreliable for some query results. It is critical for any selected resource to stop at a given time or cycle due to time-out connectivity and time-out for aggregating relevant data. Additionally, the network size represented by the number of resources at each request might high if the query is diverse. Also, system limitations between resources typically result when a small number of the active resources being selected instantly. As a result, at the moment of determining the type of query and specifying the relevant resources extremely increase the system challenges. Thus, adapted learning algorithms that are improved must therefore: (i) exploit a small amount of resources in irrelevant queries, (ii) stop slower resources in the network when time exceeded properly.

6. Conclusion

This paper proposes a novel algorithm for developing the ground-breaking research initiated at the University of Ottawa in 2012. A great deal of effort has been done for building a real-time web search algorithm for various types of queries to remedy some drawbacks in context of related approaches. Building real-time approach with real-time events is not easy task. We showed how to implement an algorithm that uses the Web's predominant application protocol to leverage REST's tenets. We explained how Wikipedia is relevant in real-time task and how is relevant for specifying types of queries and topics of resources with low disk overhead. Exploiting multiple data sources simultaneously could thereby provide end-users with real-time query results directly from the desired information resources. Query results could be integrated to look as they are from one source, or can be displayed in separated sections of the same search resulting list. However, it can be difficult to rank results from disparate resources in real-time, as this requires application of multiple predictive algorithms and understanding deep learning concepts. Our framework[6] faced many issues, including: coordination, synchronization, topic correlation, and controlling multiple autonomous web resources.

Conflict of Interest

The author declares that there is no a competing financial interest or personal relationships that could have appeared to influence the work reported in this manuscript.

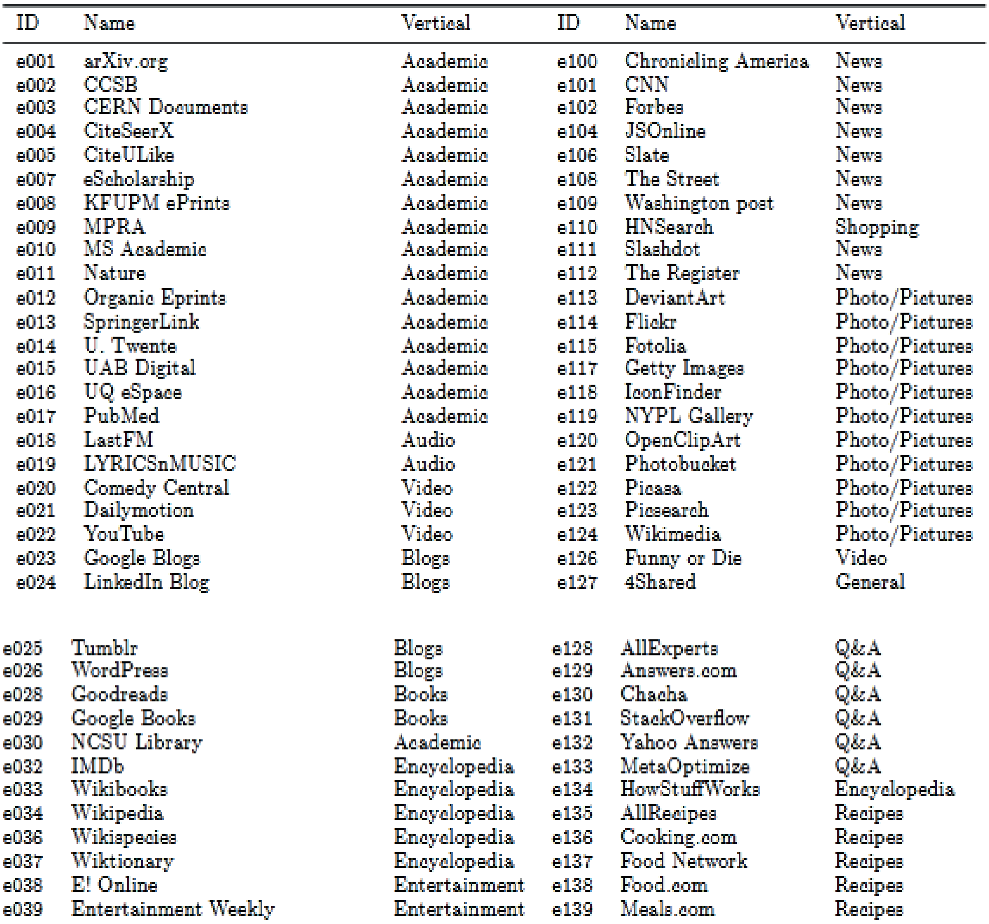
Appendix A: Real time resources used by our proposed approach

Resources	Agent / Topics	Resources	Agent / Topics
Indeed.co.uk	Jobs	Wikihow.com	Q/A
Totaljobs.com	Jobs	Mapquest.com	Local
Optnation.com	Jobs	Embassypasges.com	Local
Monster.co.uk	Jobs	Zoominfo.com	Local
Wikicfp.com	Academic	Foursquare.com	Local
Dblp.org	Academic	Openweathermap.org	Weather
Citeseerx.edu	Academic	Wikipedia.org	Encyclopedia
Univerzities.com	Academic	Nationsonline.com	Encyclopedia
Conferencealert.com	Academic	Gov.uk	Encyclopedia
Researchgate.com	Academic	Geonames.org	Encyclopedia
Taste.com	Recipes	Nationsonline.org	Encyclopedia
Simplerecipes.com	Recipes	Reference.com	Encyclopedia
Saveland.ca	Shopping	Thoughtco.com	Encyclopedia
Amazon.com	Shopping	NPR.org	Encyclopedia
Ebay.com	Shopping	News-medical.net	News
Alibaba.com	Shopping	Uptodate.com	Health
Answers.com	Q/A	Verywell.com	Health
Answers.yahoo.com	Q/A	Mayoclinic.org	Health
Drugs.com	Health	Thebalance.com	Finance
Everydayhealth.com	Health		Finance

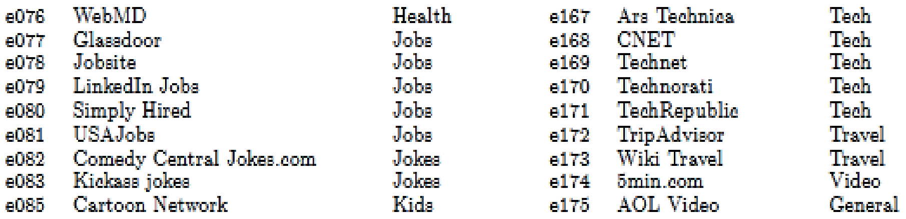
[6] http://site.uottawa.ca/~falak081

Nhs.uk	Health	Historynet.com	History
Patient.info `	Health	Twitter.com	Social
derm101.com	Health	Facebook.com	Social
Cntravel.com	Health	Wordpress.com	Blog
Hotelscombined.com	Travel	Thefreedictionary.com	Dictionary
Tripsavvy.com	Travel	Merriam-webster.com	Dictionary
Tvguide.com	Travel	Cambridge.org	Dictionary
Hollywoodlife.com	Shows	Google.com	Scholar
Imdb.com	Shows	Worldcat.org	Scholar
Thespruce.com	Shows	360daily.com	Video
Investopedia.com	Home	Vimeo.com	Video

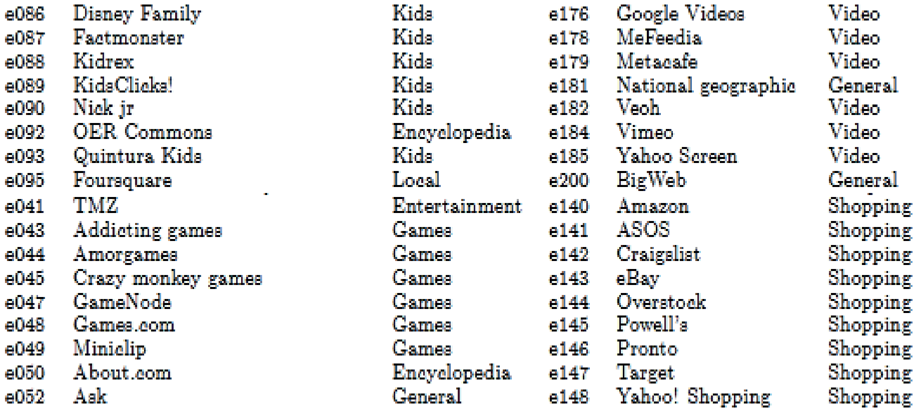
Appendix B: Real time resources used by related works



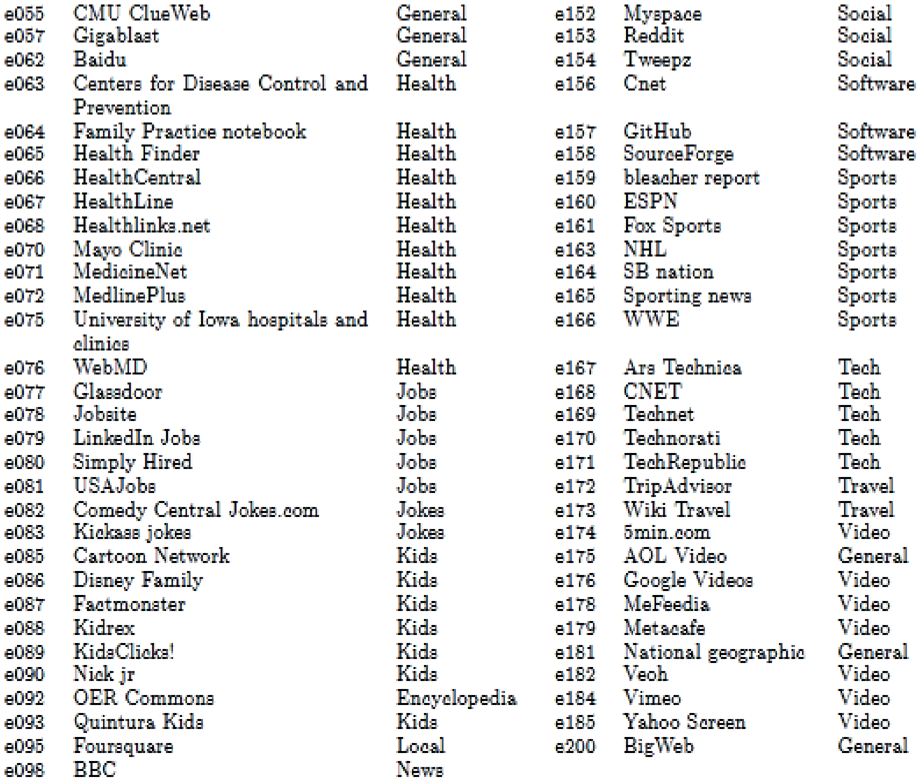
ID	Name	Vertical	ID	Name	Vertical
e001	arXiv.org	Academic	e100	Chronicling America	News
e002	CCSB	Academic	e101	CNN	News
e003	CERN Documents	Academic	e102	Forbes	News
e004	CiteSeerX	Academic	e104	JSOnline	News
e005	CiteULike	Academic	e106	Slate	News
e007	eScholarship	Academic	e108	The Street	News
e008	KFUPM ePrints	Academic	e109	Washington post	News
e009	MPRA	Academic	e110	HNSearch	Shopping
e010	MS Academic	Academic	e111	Slashdot	News
e011	Nature	Academic	e112	The Register	News
e012	Organic Eprints	Academic	e113	DeviantArt	Photo/Pictures
e013	SpringerLink	Academic	e114	Flickr	Photo/Pictures
e014	U. Twente	Academic	e115	Fotolia	Photo/Pictures
e015	UAB Digital	Academic	e117	Getty Images	Photo/Pictures
e016	UQ eSpace	Academic	e118	IconFinder	Photo/Pictures
e017	PubMed	Academic	e119	NYPL Gallery	Photo/Pictures
e018	LastFM	Audio	e120	OpenClipArt	Photo/Pictures
e019	LYRICSnMUSIC	Audio	e121	Photobucket	Photo/Pictures
e020	Comedy Central	Video	e122	Picasa	Photo/Pictures
e021	Dailymotion	Video	e123	Picsearch	Photo/Pictures
e022	YouTube	Video	e124	Wikimedia	Photo/Pictures
e023	Google Blogs	Blogs	e126	Funny or Die	Video
e024	LinkedIn Blog	Blogs	e127	4Shared	General
e025	Tumblr	Blogs	e128	AllExperts	Q&A
e026	WordPress	Blogs	e129	Answers.com	Q&A
e028	Goodreads	Books	e130	Chacha	Q&A
e029	Google Books	Books	e131	StackOverflow	Q&A
e030	NCSU Library	Academic	e132	Yahoo Answers	Q&A
e032	IMDb	Encyclopedia	e133	MetaOptimize	Q&A
e033	Wikibooks	Encyclopedia	e134	HowStuffWorks	Encyclopedia
e034	Wikipedia	Encyclopedia	e135	AllRecipes	Recipes
e036	Wikispecies	Encyclopedia	e136	Cooking.com	Recipes
e037	Wiktionary	Encyclopedia	e137	Food Network	Recipes
e038	E! Online	Entertainment	e138	Food.com	Recipes
e039	Entertainment Weekly	Entertainment	e139	Meals.com	Recipes



e076	WebMD	Health	e167	Ars Technica	Tech
e077	Glassdoor	Jobs	e168	CNET	Tech
e078	Jobsite	Jobs	e169	Technet	Tech
e079	LinkedIn Jobs	Jobs	e170	Technorati	Tech
e080	Simply Hired	Jobs	e171	TechRepublic	Tech
e081	USAJobs	Jobs	e172	TripAdvisor	Travel
e082	Comedy Central Jokes.com	Jokes	e173	Wiki Travel	Travel
e083	Kickass jokes	Jokes	e174	5min.com	Video
e085	Cartoon Network	Kids	e175	AOL Video	General

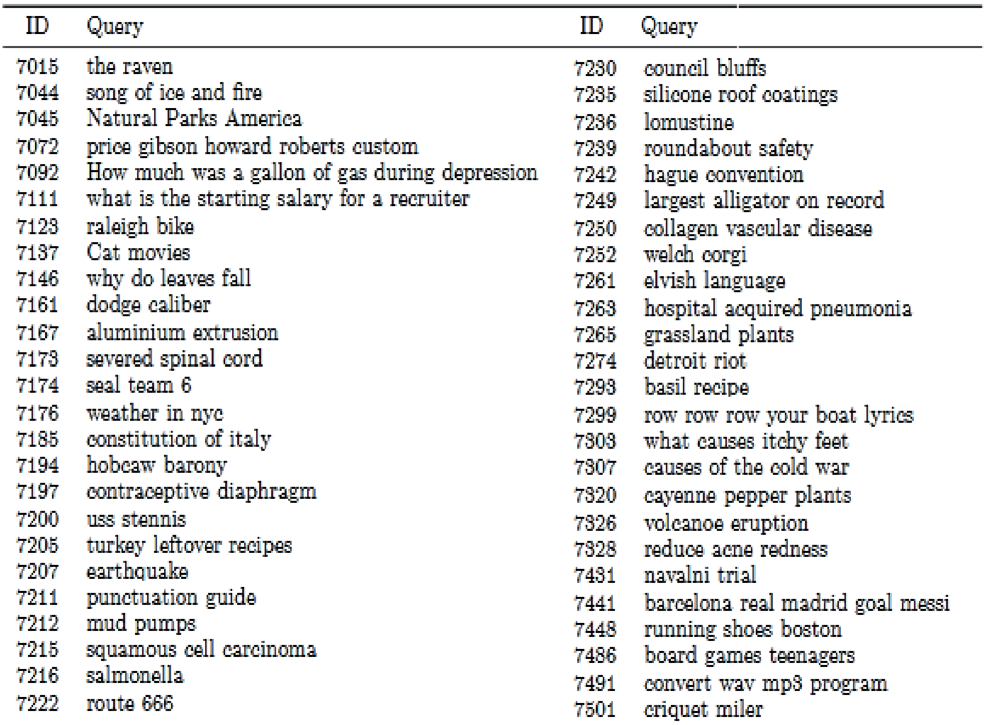


e086	Disney Family	Kids	e176	Google Videos	Video
e087	Factmonster	Kids	e178	MeFeedia	Video
e088	Kidrex	Kids	e179	Metacafe	Video
e089	KidsClicks!	Kids	e181	National geographic	General
e090	Nick jr	Kids	e182	Veoh	Video
e092	OER Commons	Encyclopedia	e184	Vimeo	Video
e093	Quintura Kids	Kids	e185	Yahoo Screen	Video
e095	Foursquare	Local	e200	BigWeb	General
e041	TMZ	Entertainment	e140	Amazon	Shopping
e043	Addicting games	Games	e141	ASOS	Shopping
e044	Amorgames	Games	e142	Craigslist	Shopping
e045	Crazy monkey games	Games	e143	eBay	Shopping
e047	GameNode	Games	e144	Overstock	Shopping
e048	Games.com	Games	e145	Powell's	Shopping
e049	Miniclip	Games	e146	Pronto	Shopping
e050	About.com	Encyclopedia	e147	Target	Shopping
e052	Ask	General	e148	Yahoo! Shopping	Shopping



e055	CMU ClueWeb	General	e152	Myspace	Social
e057	Gigablast	General	e153	Reddit	Social
e062	Baidu	General	e154	Tweepz	Social
e063	Centers for Disease Control and Prevention	Health	e156	Cnet	Software
e064	Family Practice notebook	Health	e157	GitHub	Software
e065	Health Finder	Health	e158	SourceForge	Software
e066	HealthCentral	Health	e159	bleacher report	Sports
e067	HealthLine	Health	e160	ESPN	Sports
e068	Healthlinks.net	Health	e161	Fox Sports	Sports
e070	Mayo Clinic	Health	e163	NHL	Sports
e071	MedicineNet	Health	e164	SB nation	Sports
e072	MedlinePlus	Health	e165	Sporting news	Sports
e075	University of Iowa hospitals and clinics	Health	e166	WWE	Sports
e076	WebMD	Health	e167	Ars Technica	Tech
e077	Glassdoor	Jobs	e168	CNET	Tech
e078	Jobsite	Jobs	e169	Technet	Tech
e079	LinkedIn Jobs	Jobs	e170	Technorati	Tech
e080	Simply Hired	Jobs	e171	TechRepublic	Tech
e081	USAJobs	Jobs	e172	TripAdvisor	Travel
e082	Comedy Central Jokes.com	Jokes	e173	Wiki Travel	Travel
e083	Kickass jokes	Jokes	e174	5min.com	Video
e085	Cartoon Network	Kids	e175	AOL Video	General
e086	Disney Family	Kids	e176	Google Videos	Video
e087	Factmonster	Kids	e178	MeFeedia	Video
e088	Kidrex	Kids	e179	Metacafe	Video
e089	KidsClicks!	Kids	e181	National geographic	General
e090	Nick jr	Kids	e182	Veoh	Video
e092	OER Commons	Encyclopedia	e184	Vimeo	Video
e093	Quintura Kids	Kids	e185	Yahoo Screen	Video
e095	Foursquare	Local	e200	BigWeb	General
e098	BBC	News			

Appendix C: Online Test Queries



ID	Query	ID	Query
7015	the raven	7230	council bluffs
7044	song of ice and fire	7235	silicone roof coatings
7045	Natural Parks America	7236	lomustine
7072	price gibson howard roberts custom	7239	roundabout safety
7092	How much was a gallon of gas during depression	7242	hague convention
7111	what is the starting salary for a recruiter	7249	largest alligator on record
7123	raleigh bike	7250	collagen vascular disease
7137	Cat movies	7252	welch corgi
7146	why do leaves fall	7261	elvish language
7161	dodge caliber	7263	hospital acquired pneumonia
7167	aluminium extrusion	7265	grassland plants
7173	severed spinal cord	7274	detroit riot
7174	seal team 6	7293	basil recipe
7176	weather in nyc	7299	row row row your boat lyrics
7185	constitution of italy	7303	what causes itchy feet
7194	hobcaw barony	7307	causes of the cold war
7197	contraceptive diaphragm	7320	cayenne pepper plants
7200	uss stennis	7326	volcanoe eruption
7205	turkey leftover recipes	7328	reduce acne redness
7207	earthquake	7431	navalni trial
7211	punctuation guide	7441	barcelona real madrid goal messi
7212	mud pumps	7448	running shoes boston
7215	squamous cell carcinoma	7486	board games teenagers
7216	salmonella	7491	convert wav mp3 program
7222	route 666	7501	criquet miler

Appendix D: Offline Test Queries

151:403b
152:angular cheilitis
153:pocono
154:figs
155:last supper painting
156:university of phoenix
157:the beatles rock band
158:septic system design
159:porterville
160:grilling
161:furniture for small spaces
162:dnr
163:arkansas
164:hobby stores
165:blue throated hummingbird
166:computer programming
167:Barbados
168:lipoma
169:battles in the civil war
170:scooters

176:weather strip
177:best long term care insurance
178:pork tenderloin
179:black history
180:newyork hotels
181:old coins
182:quit smoking
183:kansas city mo
184:civil right movement
185:credit report
186:unc
187:vanuatu
188:internet phone service
189:gs pay rate
190:brooks brothers clearance
191:churchill downs
192:condos in florida
193:dog clean up bags
194:designer dog breeds
195:pressure washers

171:ron howard	196:sore throat
172:becoming a paralegal	197:idaho state flower
173:hip fractures	198:indiana state fairgrounds
174:rock art	199:fybromyalgia
175:signs of a heartattack	200:ontario california airport

References

[1] Jin, S. and Lan, M. (2014). "Simple May Be Best - A Simple and Effective Method for Federated Web Search via Search Engine Impact Factor Estimation," in *Proceedings of the 23rd Text Retrieval Conference* (NIST), Special Publication 500-308.

[2] Al-Akashi, F. and Inkpen, D. (2011). "Term Impact-Based Web Page Ranking," in Proceedings of the 4th Web Intelligence, Miming, and Semantics (WIMS) International Conference, Greece.

[3] Dalton, J. and Dietz, L. (2012). "Bi-directional Linkability from Wikipedia to Documents and Back Again," in *Proceedings of the 21st Text Retrieval Conference (TREC), Knowledge Base Acceleration Track*, USA.

[4] Kamps, J., Kaptein, R., and Koolen, M. (2010). "Using Anchor Text, Spam Filtering and Wikipedia for Web Search and Entity Ranking," in *Proceedings of the 19th Text Retrieval Conference (TREC)*, USA.

[5] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004). "Swoogle: A Search and Metadata Engine for the Semantic Web," in *Proceedings of the 13th ACM Transactions on Information and Knowledge management*, 1581138741/04/0011.

[6] Lawrence, S., and Giles, C. (1999). "Accessibility of information on the web," *Journal of Intelligence, Volume 11, Issue 1, pp. 32-39*, USA.

[7] Kaptein, R., Koolen, M., and Kamps, J. (2010). "Result Diversity and Entity Ranking Experiments: Anchors, Links, Text and Wikipedia," in *Proceedings of the 19th Text Retrieval Conference (TREC)*, USA.

[8] Arya, D., Ha-Thuc, V., and Sinha, S. (2015). "Personalized Federated Search at LinkedIn," in *Proceedings of the 15th International Conference on Information and Knowledge Management*, pp. 1699-1702.

[9] Shu, W., Wang, W. and Wang, Y. (2014). "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing," *EURASIP Journal on Wireless Communications and Networking*. 64 (2014). https://doi.org/10.1186/1687-1499-2014-64

[10] Schuhknecht, F., Dittrich, J., and Linden, L. (2018). “Adaptive Indexing,” in *Proceedings of 34th IEEE International Conference on Data Engineering (ICDE)*, Paris, France, April 16-19.

[11] Chen, C., Li, F., Ooi, B. C. and Wu, S. (2011), "TI: An efficient indexing mechanism for real--time search on tweets," in *Proceedings of the ACM SIGMOD, International Conference on Management of Data*, pp. 649—660.

[12] Lu J. (2007). “*Full-Text Federated Search in Peer-to-Peer Networks*,” A Ph.D. Thesis, Language Technologies Institute School of Computer Science, Carnegie Mellon University.

[13] Vasco P. (2009). “*Federated Ontology Search*,” A Ph.D. Thesis, Language Technologies Institute School of Computer Science, Carnegie Mellon University.

[14] Nguyen, D., Demeester, T., Trieschnigg, D., and Hiemstra, D. (2007). Resource Selection for Federated Search on the Web. ArXiv:1609.04556, CTIT Technical Report TR-CTIT-16-12.

[15] Wang, S., Tuor, T., Salonidis, T., Leung, K., Makaya, C., He, T., and Chan, K. (2019). “Adaptive Federated Learning in Resource Constrained Edge Computing Systems,” Journal on Selected Areas in Communications, 37:1205–1221.

[16] Li, T., Sahu, A., Talwalkar, A., and Smith, V. (2019). Federated Learning: Challenges, Methods, and Future Directions. arXiv preprint arXiv:1908.07873.

[17] Rijke, M., Kenter, T., Vries, A., Zhai, C., Jong, F., Radinsky, K., and Hofmann K. (2014). “Advances in Information Retrieval,” *36th European Conference on IR Research, ECIR*, pp. 184-196.

[18] Lee W. (2007). “Personalizing Internet Information Services: Passive Filtering and Active Retrieval,” *International Journal of Computers and Applications*, 29:2, pp. 124-131, DOI: 10.1080/1206212X.2007.11441840.

[19] Kim J. (2001). “Hybrid Filtering of web Pages for a Recommendation Agent,” *International Journal of Computers and Applications*, 23:2, pp. 99-105, DOI: 10.1080/1206212X.2001.11441638.

[20] Ponnuswami, A., Pattabiraman, K., Wu, Q., Bachrach, R., and Kanungo, T. (2011). “On composition of a federated web search result page: using online users to provide pairwise preference for heterogeneous verticals,” *Proceedings of the fourth ACM International Conference on Web Search and Data Mining* (WSDM), pp.715-724, DOI: 10.1145/1935826.1935922.

[21] Jansen, J., Booth, D., and Spink, A. (2008). “Determining the Informational, Navigational, and Transactional Intent of Web Queries,”

Information Processing & Management, 44(3):1251-1266, doi:10.1016/j.ipm.2007.07.015.

[22] Al-akashi, F. and Inkpen, D. (2012). "Intelligent Web Page Retrieval Using Wikipedia Knowledge," in *Proceedings of the 2nd Web Intelligence, Mining and Semantics (WIMS) International Conference*, Romania.

[23] MacKinnon, I., and Vechtomova, O. (2008). "Improving Complex Interactive Question Answering Enhanced with Wikipedia Anchor Text," in *Proceedings of the 30th European Conference on Advances in Information Retrieval (ECIR)*, Glasgow, UK, pp. 438-445.

[24] Xing, Y., and James, A. (2010). "A Content based Approach for Discovering Missing Anchor Text for Web Search," in *Proceedings of the 10th SIGIR*, pp. 19–23.

[25] Kekäläinen, J, and Järvelin, K. (2013). "Using graded relevance assessments in IR evaluation," in *Proceedings of the ACM SIGIR*, JASIST 53, USA.

[26] Baykan, E., Henzinger, M., and Marian, L., (2009). "Weber-l. Purely URL-based Topic Classification," in *Proceedings of the 18th International conference on World wide web (WWW)*, pp. 1109-1110, Spain.

[27] Limsopatham, N., McCreadie, R., Albakour, M., Macdonald, C., Santos, R., and Ounis, I. (2012). "University of Glasgow at TREC 2012: Experiments with Terrier in Medical Records, Microblog, and Web Tracks," in *Proceedings of the 21st TREC Web Track Conference*, USA.

[28] Xue, Y., Guo, S., Guan, F., Yu, X., Liu, Y., Cheng, X. and Li, H. (2012). "ICTNET at Web Track 2012 Ad-hoc Task". Text Retrieval Conference.

[29] Demeester, T., Trieschnig, D., Nguyen, D., Zhou, K., and Hiemstra, D. "Overview of the TREC 2014 Federated Web Search Track," The Twenty-Third Text Retrieval Conference (TREC), Proceedings, NIST Special-Publication: SP 500-308, 2014.

[30] Guan, F., Zhang, S., Liu, C., Yu, X., Liu, Y., and Cheng, X. (2014). "ICTNET at Federated Web Search Track2014", Text Retrieval Conference.

[31] Sanjulián, C. (2008). "Automatic query expansion and word sense disambiguation with long and short queries using WordNet under vector model," *Actas de los Talleres de las Jornadas de Ingeniería del Software Bases de Datos*, Vol. 2.