WikiQuery -- An Interactive Collaboration Interface for Creating, Storing and Sharing Effective CNF Queries

Le Zhao
Carnegie Mellon University
lezhao@cs.cmu.edu

Xiaozhong Liu
Indiana University, Bloomington
liu237@indiana.edu

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

ABSTRACT

Boolean Conjunctive Normal Form (CNF) expansion can effectively address the vocabulary mismatch problem, a problem that current retrieval techniques have very limited ability to solve. Meanwhile, expert searchers are found to spend large amounts of time carefully creating manual CNF queries. These CNF queries are highly effective, and can outperform bag of word queries by a large margin. However, not many effective tools exist that can facilitate the efficient manual creation of effective CNF queries.

We describe such a publicly available search tool, WikiQuery, which can efficiently assist the users to create CNF queries through easy query editing and immediate access to search results. Experiments show that ordinary search users, with limited prior knowledge of Boolean queries, can use this intuitive tool to create effective CNF queries. We argue that tools like WikiQuery can attract and retain certain users from the commercial Web search engines, and may be a good starting point to build a research Web search engine.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]:

General Terms

Theory, Experimentation, Measurement

Keywords

Wiki for queries, conjunctive normal form (CNF) queries, query refinement, user interactions

1. INTRODUCTION

One particular goal of the Open Source Information Retrieval workshop is to build 'an open source, live and functioning, online web search engine for research purposes'. A key factor necessary for the success of such an effort is to *attract* and *retain* users.

In order to attract users, the search engine needs to have a *distinct* and *useful* feature that is not offered by the current search engines. As a somewhat negative example, the Lemur community query log project did not collect enough query log data perhaps due to the lack of any additional benefit provided by the query log toolbar[1]. Compared to the toolbar, a full scale open source search engine is even more likely to fail, as the quality of the results from such an academic search engine is likely to be much worse than that from the commercial Web search engines.

In order to retain users, it is perhaps necessary that the distinct feature is *unlikely to be copied* by the competitors (the commercial Web search engines).

This paper describes one such publicly available open source search tool, WikiQuery (http://www.wikiquery.org), which both engages ordinary searchers in effective search interactions, and is unlikely to be adopted by the commercial Web search engines. WikiQuery can provide more effective search interactions than what the current search engines can offer, and is flexible enough to be applied on top of virtually any Web search engine.

Prior research showed that the current retrieval techniques are still very limited in their ability to solve the vocabulary mismatch problem [13]. Users are still frequently frustrated by the current search engines when performing informational searches [5]. Prior research also indicated that high quality manually created Conjunctive Normal Form (CNF) queries offer the opportunity to address this limitation and significantly improve retrieval beyond the traditional bag of word queries [14]. A huge potential of improvement is possible in the scale of 50-300% with carefully manually created CNF queries [14].

The WikiQuery interface is designed to guide and facilitate users to create highly effective CNF queries efficiently through 1) a simple CNF input interface, 2) immediate inspection and interaction with search results from multiple commercial search engines, and 3) collaboration with other users who share related information needs. The created queries are stored, and readily available for future re-finding or refining. The queries are also shared online so that other users may benefit from the queries or query parts. Being a Wiki website, different users can collaborate and improve queries together. This interface is implemented based on the MediaWiki source code, which allows the users to search for pages or information stored on the website, so that it is easy to lookup, share or collaborate on the website.

User studies in this work show that ordinary search users with limited knowledge of Boolean queries have the potential to use the WikiQuery interface to create effective CNF queries.

WikiQuery has the potential to attract and retain users for two reasons. Firstly, the CNF query interface is effective and intuitive, and can appeal to the ordinary search users -- at least the early adopters who are willing to learn a new and effective way to formulate search queries, or the more serious users who care about their searches. Secondly, the commercial Web search engines are very unlikely to adopt the CNF interface, because the change in user experience is large enough to scare away the change-averse users, making it very risky to use for a large Web search engine.

In addition to the added benefit of facilitating search interactions, the resulting crowdsourced CNF queries stored on the WikiQuery website also constitute a detailed context dependent thesaurus for retrieval and other vocabulary tasks.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 describes the WikiQuery website together with its CNF query interface. Section 4 reports the studies showing that ordinary users can create effective CNF queries with the proper tool and guidance. Section 5 concludes the paper.

The copyright of this article remains with the authors.
SIGIR 2012 Workshop on Open Source Information Retrieval.
August 16, 2012, Portland, Oregon, USA.

[1] http://lemurstudy.cs.umass.edu/

2. RELATED WORK

This section reviews prior work related to three aspects of this work, the uses of Boolean CNF queries, Boolean user interfaces, and the use of the user generated Boolean queries as a resource for thesaurus building. We also discuss how this research is different from prior efforts.

2.1 Uses of CNF Queries

Prior research on effective uses and formulations of Boolean CNF queries motivates this research. The use of Conjunctive Normal Form (CNF) queries is widespread among librarians [9,6], lawyers [3,2], and other expert searchers [7,4,11].

For example, the query below from TREC 2006 Legal Track [2]

"*sales of tobacco to children*"

is expanded manually into the Boolean CNF query

(*sales* OR *sell* OR *sold*) AND
(*tobacco* OR *cigar* OR *cigarettes*) AND
(*children* OR *child* OR *teen* OR *juvenile* OR *kid* OR *adolescent*)

In the above case, each query term is expanded into one conjunct of the Conjunctive Normal Form query.

Earlier research on Boolean queries examined *unranked* Boolean retrieval, and showed that ranked keyword retrieval is more effective, mainly because presenting retrieval results as a set is both difficult to control and inefficient to examine. Later research compared *ranked* Boolean with keyword retrieval, showing that user created CNF queries can significantly improve over keyword retrieval by simply grouping the query terms of the verbose keyword queries into Conjunctive Normal Form [7,11].

More recent research showed that lawyers and search experts can create highly effective CNF queries that extensively expand the original keyword queries, solving mismatch and improving retrieval 50-300% [14]. These CNF queries with high quality expansion terms were shown to outperform bag of word expansion with the same set of high quality expansion terms.

2.2 Boolean Search Interfaces

Even though carefully created CNF queries are effective, recent research has focused on bag of word queries, and has not seen much development in interfaces that help users create effective CNF queries. Research on Boolean user interfaces happened mostly before mid 1990s. Hearst [1, Chapter 10] cited several textual as well as graphical Boolean interfaces. Hearst referred to CNF queries as *faceted queries*, and described a possible textual input interface for CNF queries, though without a concrete example. In a newer book, Hearst [8] cited the advanced search interface of the Educational Resources Information Center (ERIC)[2], which allows the entry of CNF queries in a one-conjunct-per-line format. This is similar to the CNF interface of WikiQuery except for two differences. Firstly, the ERIC interface is not specifically designed for CNF queries, and allows the user to enter a query in Disjunctive Normal Form. Secondly, the ERIC interface gives no guidance or useful examples to the user on how to create effective Boolean queries.

The lack of research on Boolean interfaces is coupled with a long list of negative results [8, Section 4.4] showing that ordinary users have a difficult time formulating effective Boolean queries. This work, on the contrary, shows that ordinary search users with limited knowledge of Boolean queries have the potential to create effective Boolean CNF queries using the WikiQuery interface. This apparent contradiction is likely because the prior studies did not focus on Boolean CNF queries, and gave novice users the full freedom of free form Boolean queries without proper guidance. This choice leaves the creation of effective Boolean queries to the chances, and is likely to lead to ineffective Boolean queries. Our results point at a promising direction of designing search interfaces that guide and facilitate users to formulate effective Boolean queries in CNF form.

[2] http://www.eric.ed.gov/ERICWebPortal/search/extended.jsp accessed on June 1st, 2012.

2.3 Online Thesaurus Building

The resulting CNF queries created by users and stored in WikiQuery can serve as a thesaurus for future users. In particular, each conjunct in the CNF queries contains synonyms or related terms that are dependent on the context of the query. Compared to existing thesauri like WordNet, the WikiQuery synonyms depend on the specific uses of a term in a query, while WordNet is still a static semantic resource without regard to word use.

The thesaurus building aspect of the WikiQuery website is similar to an earlier system that builds a growing thesaurus based on users' Boolean retrieval interactions [12]. The main difference is the emphasis on CNF queries by WikiQuery. WikiQuery also treats individual queries as valuable resources, and as units for storage and retrieval. This is a fairly lazy and ad hoc treatment for a thesaurus. Later more general treatments can build on top of the queries stored on WikiQuery, when it becomes clear what kinds of general treatments are most appropriate.

3. THE WIKIQUERY WEBSITE

The search tool described in this work is a public Wiki website based on the same source code that supports Wikipedia etc. sites.

On the WikiQuery website, each Wiki page stores all the information about one particular user information need, including possibly a description of the information need, the corresponding CNF query (or several related CNF queries), possible relevant results (together with descriptions) identified through the search interactions, or other related information.

An example WikiQuery page is shown in Figure 1. The main CNF query of the page and the links to the search engine result pages from multiple search engines are circled out.

The open source MediaWiki code (http://www.mediawiki.org) offers the standard set of features used in popular Wiki websites. One useful function allows the users to search for pages or information stored on the website through entering a search query in the search box. In addition, being based on MediaWiki version 1.17, the Wiki website automatically suggests existing WikiQuery pages as the user types into the search box. Other features include history tracking of all the user edits of pages and users, subscribing to a page to monitor changes made to the page, and opportunity of discussion among contributors of a Wiki page.

Several simple customizations were made to accommodate the special user needs for the WikiQuery website, including 1) a simple textual interface for CNF query editing, 2) an automatic client side script to display the query and store it in the Wiki page, and 3) an automatic script that translates the CNF query into the formats accepted by common Web search engines, allowing immediate inspection of the retrieval results produced by the CNF queries. The rest of this section covers these customizations in more detail.

3.1 Interface for CNF Query Editing

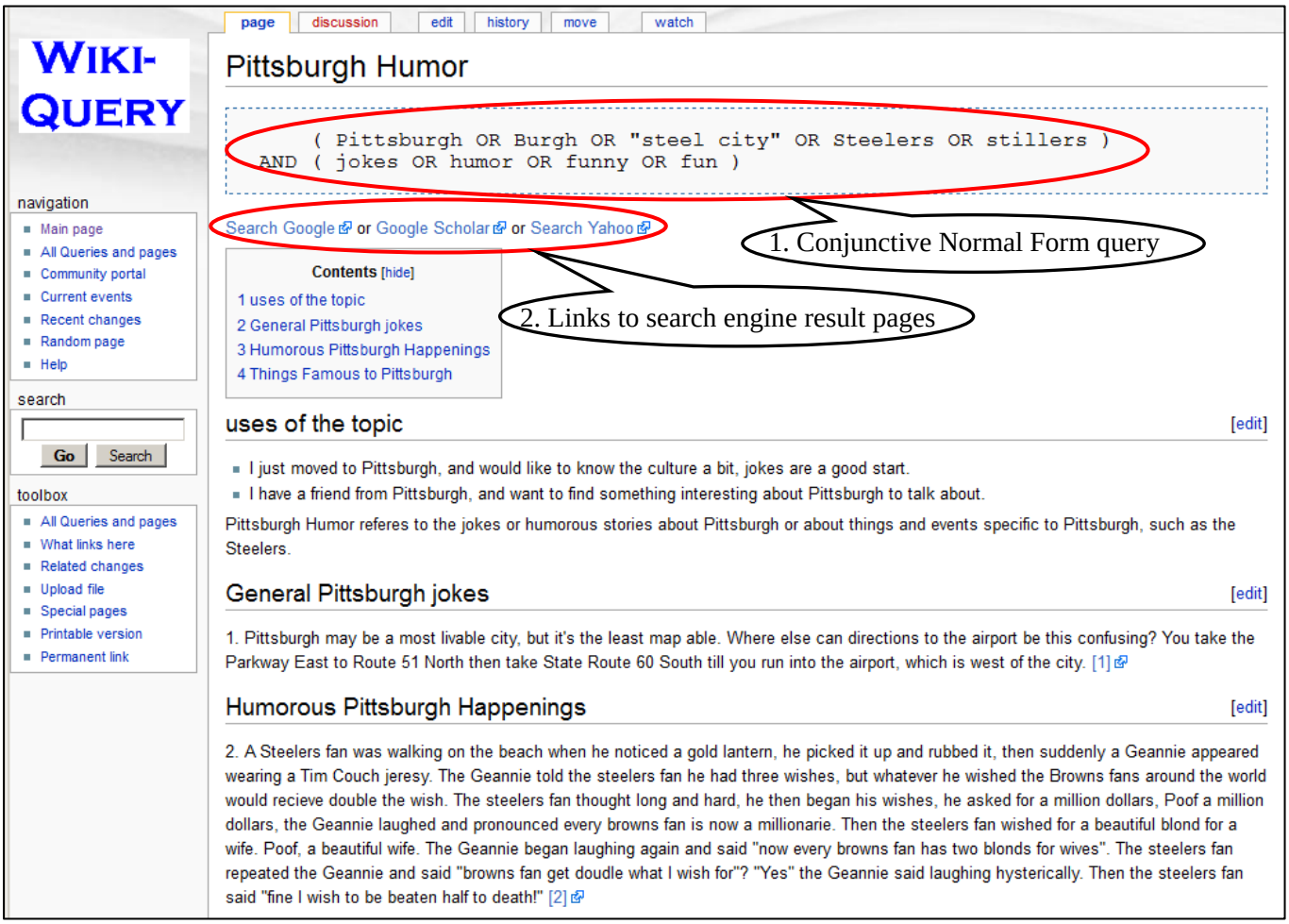

Figure 1. An example Wiki page from the WikiQuery Website. Circled out are the Conjunctive Normal Form (CNF) query of this Wiki page and the links to the search engine result pages for the CNF query.

The CNF query interface is presented to the user when the user edits a Wiki page. It guides the user and allows the user to easily and efficiently create or edit CNF queries. As shown in Figure 2, this interface is consisted of several input bars, each corresponding to one conjunct in the CNF query. The user has to determine how many and what concepts (conjuncts) are necessary for the particular information need. Then the user has to enter in each input bar the search terms that can be used to describe the concept, and join them with the Boolean OR operator. Prior research [14] indicated that including more high quality expansion terms in each conjunct yields a higher likelihood for the conjunct to match the relevant documents of the query, and leads to a higher retrieval accuracy.

The CNF queries are stored on the wiki to allow users to revisit existing queries, and to further improve the queries. Because refinding tasks are fairly common in Web search, a user might frequently find the stored queries to be helpful at a future time.

Whenever the user edits an existing WikiQuery page that already contains a full CNF query, the CNF input bars are automatically populated with the content of the CNF query, so that the user does not have to enter the query into the input bars again.

The collaborative nature of the Wiki website also allows different users to collaborate and edit the same WikiQuery page of common interest to these users. For popular information needs, collaborations across multiple users are likely to improve the quality of the CNF queries beyond what a single user may achieve. Because high quality CNF queries can take lots of effort to create, collaboration offers the possibility to break down the difficulty through sharing it among a group of users.

This Boolean CNF interface is different from the typical interfaces used in advanced searches in libraries or by lawyers in legal discovery. The advanced search interfaces in libraries (e.g. Library of Congress) allow a restricted Boolean query of the form: Term1 Op1 Term2 Op2 …, where a Term can be a word or a phrase, but *cannot* be a Boolean clause, and an Op is a Boolean operator (e.g. AND, OR, XOR, and NOT). The WikiQuery CNF interface is more powerful than the library Boolean interfaces because any Boolean query can be expressed as a CNF query.

The Boolean interface used by lawyers is usually just one large text box. They are flexible and allow free form Boolean queries to be entered into a, typically large, text box. However, the lawyers typically create CNF-like queries [2], and have to enter the whole query by themselves, having to make sure that the parentheses match and the form is correct. The WikiQuery CNF interface facilitates simpler and more efficient manual creation of CNF-like queries. It breaks down each query by allowing the user to enter each conjunct into one input box. This way, the user does not need to enter the CNF skeleton, nor the conjunct level parentheses. Although this CNF interface suggests the use of CNF-like queries, it does not require that. The user still has the freedom to

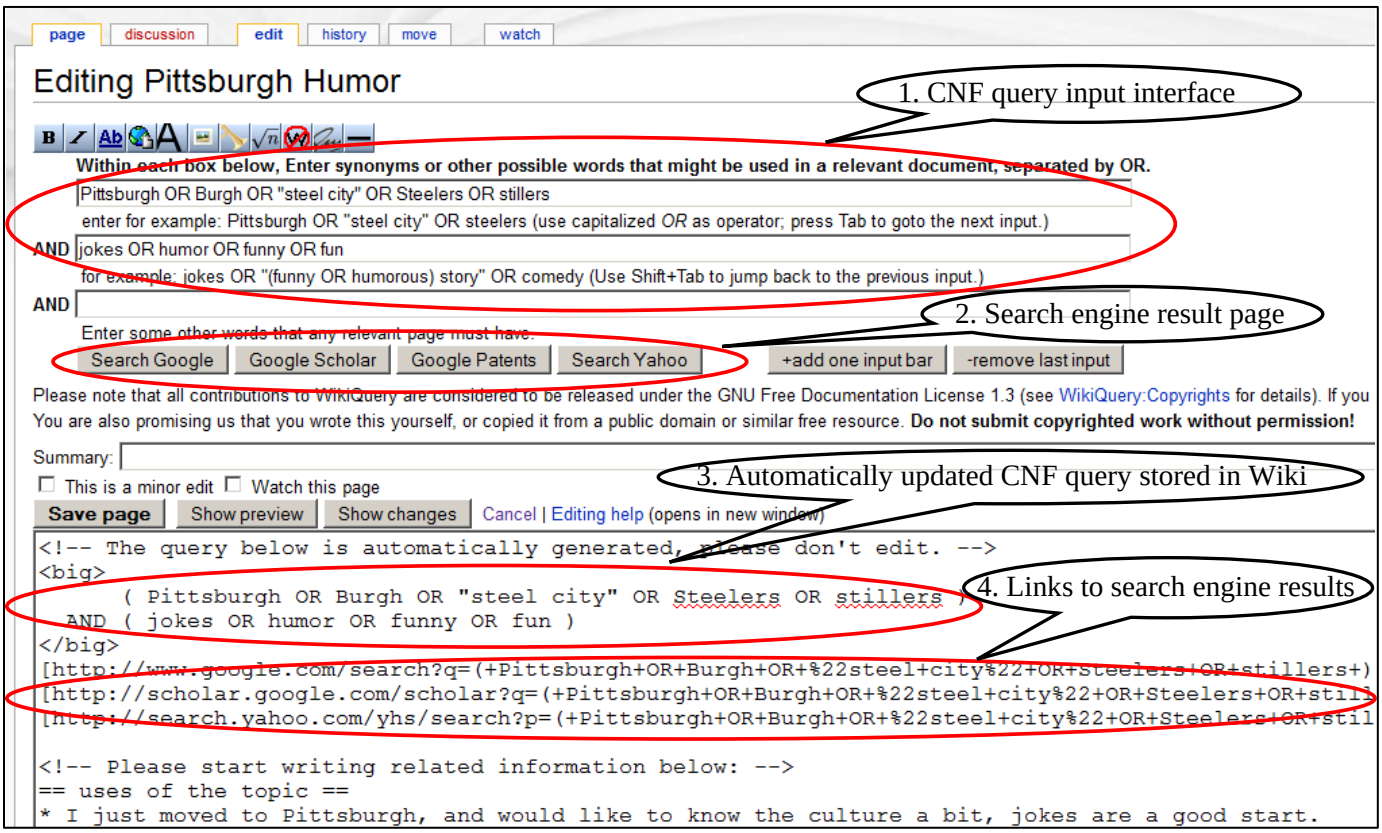

Figure 2. The editing interface of WikiQuery. It includes a Conjunctive Normal Form (CNF) query editing interface and buttons to access search engine result pages. The full CNF query and the HTTP links to the search engine result pages for the query are automatically generated and stored in the page.

enter a free form Boolean query into one input bar in the WikiQuery interface, although this usage is generally discouraged.

3.2 Query Storage and Display

The query that the user enters into the CNF editing interface (Figure 2) is automatically translated into the full form CNF query that is stored and displayed on each Wiki page (as seen in Figure 1). Whenever the user makes a change in one of the input bars in the CNF edit interface, a short Javascript is automatically triggered to translate the contents of the input bars into the full query as part of the content of the Wiki page to be stored.

Because the editing interface uses the document content of the Wiki page to store and display CNF queries, all the benefits from MediaWiki for maintaining the Wiki contents automatically apply to the stored CNF queries, change tracking and searching.

3.3 Querying Search Engines

The WikiQuery website allows the user to access search engine result pages for the CNF queries very easily, during page viewing and editing. Access to the result pages during page view allows viewers of the WikiQuery website to easily check the search engine results for a CNF query. Access to the result pages in the editing interface allows the user to monitor the quality of the search engine results after making changes to the CNF queries, ensuring the quality of the resulting CNF queries.

Figure 1 shows the links displayed on the stored WikiQuery pages during page viewing. Figure 2 shows the buttons that would open a new window to allow the user to navigate to the search engine result pages for the query that the user is editing.

Multiple search engines are supported. These CNF queries can work on any search engine that supports Boolean query operators. Currently it employs Google, Google Scholar, Google Patents and Yahoo (Altavista), but can be easily extended to use other search engines like Bing which uses a slightly different query language.

4. USER STUDY

Prior research already confirmed the effectiveness of the manual CNF queries, and that expert searchers can create effective CNF queries [14]. The study in this section aims to verify the hypothesis that ordinary users with no or limited prior knowledge of Boolean queries can create effective Boolean CNF queries using the WikiQuery CNF interface.

6 users participated in this preliminary user study, each responsible for 2 information needs. Users typically proposed a series of Boolean queries. We report the retrieval performance of the Boolean queries against the baseline keyword queries.

4.1 Experiment Setup

This subsection describes the details of this experiment, including user selection, information needs, evaluation details, relevance judgments and evaluation methodology.

4.1.1 Classroom Users

Users for this study come from an IR class in an information school. A total of 6 students participated in the study. We purposely launched the study at the beginning of the semester when students were not fully exposed to the professional CNF queries.

As participants had little knowledge of Boolean queries or Wiki page editing, a 10-minute session was given before the study,

which included an example information need with a walk through the CNF query creation and editing process on WikiQuery. Because this study counts as one homework assignment for the students, participants were highly motivated to spend time and do well in creating effective CNF queries. Users were also asked to document the detailed query formulation and retrieval experience.

4.1.2 Information Needs

12 topics from TREC Ad hoc and Terabyte tracks were selected as candidate topics. Topics were selected to be somewhat interesting for the students, and reasonably difficult so that the keyword queries were unlikely to return perfect results. Each student was randomly assigned two topics, for which the student would assume the role of the searcher and create queries.

Each TREC topic contains a short title and a long description of the information need. The students used all the available information of the topic to grasp the intent behind the topic. They generated queries for each topic and were encouraged to interact with the search results to improve the proposed queries.

One reason for using standard TREC topics is to use the existing relevance judgments on these datasets to evaluate the user queries.

4.1.3 Baseline Keyword Queries

The keyword queries were directly taken from the TREC topic titles and descriptions. The topic titles are shorter, usually 2 to 4 terms long, and the descriptions are much longer, around 5 to 10 terms long. These two types of keyword queries correspond to the two baselines: keyword *title* and keyword *desc*.

4.1.4 User Created Boolean Queries

Users were asked to create Boolean CNF queries using the WikiQuery interface, which allows them to enter the query, to examine the results returned by the search engines for the query, and to improve the query. On average, a user spent around 40 minutes on each information need, based on the recorded history of changes of the WikiQuery pages. For each information need, the users created several queries or improved the CNF query many times.

The users were asked to submit two versions of Boolean queries for each information need, an initial Boolean query and a final version of the Boolean query. The initial Boolean query represents a very first try by the user, and the final Boolean query is usually the result of fine tuning the CNF query based on interactions with the retrieval results of all the queries the user tested.

4.1.5 Evaluating on TREC Datasets

Since the information needs come from official TREC Ad hoc track and Terabyte track topics, existing relevance judgments from TREC can be used to evaluate the effectiveness of the user proposed queries. The advantage of using the TREC relevance judgments is that these judgments are fairly reusable, and more complete than just evaluating several top results returned by a search engine. Thus, one can evaluate the result lists returned on the TREC datasets to much deeper levels. TREC standard evaluation metrics report retrieval accuracy of the top 1000 results. One may question why such deep level of assessment is necessary for Web search where users typically only look at top several results. We argue that the deeper level metrics are more sensitive to retrieval algorithm differences. Certain retrieval algorithm changes may not surface as top rank result changes on a small set of test topics, but may change the rank list more dramatically at deeper levels for these topics. These deeper level changes may show up at the top ranks on a small subset of topics of a much larger test topic set. At the very least, the deeper level metrics provide an additional perspective to the effectiveness of the rank lists.

The TREC relevance judgments only exist on the TREC document collections, so the queries need to be run against the smaller TREC collections, instead of a Web search engine.

We used the Indri search engine of the Lemur toolkit version 4.10 to execute the Boolean queries on TREC document collections. Indri supports a fairly comprehensive query language. The backend model is language model with Dirichlet smoothing. The Boolean OR operator is implemented as the Indri *#syn* operator. #syn counts term frequency and document frequency of the whole group of synonyms by treating all the synonyms in the group as the same term. The Boolean AND operator is implemented as the Indri *#combine* operator which is the probabilistic AND operator. This Indri implementation of a Boolean query automatically returns a rank list of documents, instead of an unranked set. This is a more effective form of result presentation than an unranked set of documents, and is widely adopted by modern retrieval systems.

Equations (1, 2) show how Indri scores document d with query (a OR b) AND (c OR e). tf(a, d) is the number of times term a appears in document d. μ is the parameter for Dirichlet smoothing, which is set at 900 for the Ad hoc track datasets and 1500 for the Terabyte track datasets.

$$\begin{aligned} &\text{Score}(\ (a \text{ OR } b) \text{ AND } (c \text{ OR } e),\ d) \qquad (1)\\ &= P(\ (a \text{ OR } b) \text{ AND } (c \text{ OR } e) \mid d)\\ &= P(\ (a \text{ OR } b) \mid d) * P(\ (c \text{ OR } e) \mid d) \end{aligned}$$

$$\begin{aligned} &P(\ (a \text{ OR } b) \mid d) \qquad (2)\\ &= (\ \text{tf}(a, d) + \text{tf}(b, d) + \mu * (P(a \mid C) + P(b \mid C))\ ) / (\text{length}(d) + \mu)\\ &= P(a \mid d) + P(b \mid d) \qquad \text{(under Dirichlet smoothing)} \end{aligned}$$

4.1.6 Evaluating on Commercial Search Engines

The WikiQuery website allows the users to run the CNF queries they created on commercial Web search engines, which typically have a much larger collection of documents, and the retrieval algorithms are typically more effective than the experimental systems used by researchers. This section tries to evaluate the effectiveness of the CNF queries by running them on commercial search engines.

Given a Boolean CNF query as input, the Web search engines return ranked lists of documents as results. The exact ranking formulae of these search engines are difficult to know. Standard ways of producing ranked retrieval results from Boolean queries include probabilistic Boolean retrieval models, quorum-level ranking [8, Section 4.4.2], or simply using keyword retrieval to rank the documents that match the Boolean query. An empirical comparison of some of them can be found in [14].

Relevance judgments are needed to evaluate the effectiveness of the results returned by the search engines. Users who participated in the study did relevance judgments for each other for the top 5 results returned by the Web search engines (Google and Yahoo). The assessors *did not know* what query, search engine or rank a particular result page comes from. These user-provided judgments were obtained at the time of the homework assignment (February to March 2011). One of the authors of this paper verified these user provided relevance judgments for accuracy.

4.1.7 Evaluation Metrics

For TREC judgments, we report Mean Average Precision (MAP) for the top 1000 results. It is a standard measure because it is sensitive to rank list changes and also fairly stable when comparing between systems [10]. We also report Precision at top 5 and 10 as measures of retrieval accuracy at top ranks. For evaluation on the search engines, we report Precision at top 5, and MAP at 5, as deeper relevance judgments are not available.

Table 1. The differences between short keyword and the final Boolean CNF query for each TREC topic. The **types of changes** include *restricting* the query (by including more conjuncts or using phrase operator to require query terms to occur close together) and *expanding* the query terms by including more synonyms (highlighted by shading the topic number).

Topic No./ Type of Change	Query with changes from *short keyword* to *final Boolean*, (**bolded** are insertions and ~~slashed~~ are removals).
352 both	#combine(#syn(**#1(**British Chunnel**) #1(Channel Tunnel)**) **#syn(#1(effect on) changes)** #syn(**#1(**#syn(British **UK**) **economy) #1(economic #syn(implications changes evaluation))**) ~~impact~~)
354 expand	#combine(#syn(**reporter newswriter** journalist **correspondent**) **#syn(arrested hostage #1(physical attack) killed threatened kidnapped murdered attack shot)** ~~risks~~)
704 restrict	#combine(**#1(**green party**) #syn(US #1(united states))** #syn(**#1(**political views**) politics**))
751 expand	#combine(scrabble #syn(players **group**) #syn(social **events**))
752 restrict	#combine(**#syn(location places countries)** dam removal Environmental impact **reason**)
753 restrict	#combine(bullying prevention programs **in schools #syn(classes assemblies discipline mediation projects) #syn(Students staff)**)
758 expand	#combine(embryonic stem cells #syn(restrictions **law policy**))
760 both	#combine(statistics **#1(in** America**)** Muslims **#syn(population demographics)** #syn(mosque **#band(Islamic center)**) school)
764 restrict	#combine(measures improve public transportation ~~increase mass transit use~~)
769 restrict	#combine(**#1(**Kroll Associates**)** employee names)
799 restrict	#combine(**type of animals** Alzheimer research))
805 restrict	#combine(identity theft passport **help victims identify establish credit worthiness show #syn(creditors #band(law enforcement))**)

Note: #combine is Indri's probabilistic AND operator, #syn is Boolean OR, #1 is the phrase operator, and #band is the Boolean AND.

4.2 Experiment Results

This section reports the characteristics of the user generated CNF queries, and the effectiveness of these CNF queries compared against the keyword query baselines.

4.2.1 Characteristics of the User CNF Queries

Query characteristics varied a lot across different users: 2 to 6 conjuncts for the *initial* Boolean queries, each conjunct containing 1 to 5 synonyms. The *final* Boolean queries were expanded a bit more, with 2 to 6 conjuncts, each containing 1 to 9 synonyms.

Table 1 shows how the users modified the original short keyword queries into the final Boolean CNF queries.

Users did not always follow the instruction to include expansion terms when formulating CNF queries. Only 5 of the 12 queries included some synonym expansion, while 9 out of the 12 queries were modified to be more restrictive than the keyword query. These queries are less well expanded than the queries created by expert searchers [2,14]. The sections below show how that affects retrieval performance.

4.2.2 Effectiveness – TREC Evaluation

Retrieving on the TREC datasets, the user created CNF queries are fairly effective overall (Table 2). On average, the final Boolean CNF queries perform the best on all three evaluation metrics. These final CNF queries perform significantly better than the long keyword queries both at top ranks (P@5, 10) and at overall accuracy (MAP@1000), and outperform on average the short keyword queries. This result is consistent with prior research, where shorter keyword queries perform better than long queries [13].

Even though CNF queries are better than short keyword queries, the difference is not statistically significant. We look at the individual queries to understand which CNF queries are better and which are worse than the corresponding short keyword query.

For expansion queries in Table 3 (topics 354, 751 and 758), topic 354 is the only one that decreases performance. The reason is that "reporter" is stemmed and matches the word "report", which is a common word in the TREC newswire collection, causing the expanded query to match many false positives.

For the restrictive Boolean queries, the performance gain is less stable. 5 (topics 752, 760, 764, 799, 805) out of the 9 restrictive Boolean queries perform worse than the short keyword query in MAP. Even in top precision, which is usually the users' goal for using more restrictive queries, 4 out of the 9 restrictive Boolean queries perform worse than short keyword queries.

This result on TREC datasets shows that many of the CNF expansion queries and a few of the restrictive Boolean queries created through interacting with a larger dataset (the Web) and very different retrieval algorithms can still effectively retrieve relevant documents on a smaller dataset. For expansion queries, this may be because on the one hand, accurate CNF expansions on larger collections are less likely to match false positives on smaller collections, ensuring precision. On the other hand, the mismatch problem is likely to get worse on the smaller collections with fewer relevant documents, thus, the CNF expansions are more likely to be useful in improving recall on the smaller collections. Overall, in both precision and recall, the CNF expansions created for larger collections may work well on the smaller collections. However, the more restrictive queries that perform well on large Web collections may not perform as well on smaller collections.

4.2.3 Effectiveness – Evaluation on Search Engines

The Boolean queries are clearly more effective than short keywords on the commercial Web search engines at top ranks, as shown in Table 4 (overall) and Table 5 (per topic).

Table 4 shows the evaluation on Google and Yahoo. On Google, final CNF queries significantly outperformed the short keyword queries in retrieval performance at top ranks. On Yahoo, CNF was on average better than short keyword, but the difference was not statistically significant, because 3 Boolean queries were worse than the short keyword baseline. (For topic 352 on Yahoo, the expansion phrase "channel tunnel" is too general and matches many false positives. For topic 354 on Yahoo, many false positives contain "Reporter", "Journalist" and "Correspondent" as titles of publications or newspapers, instead of as a person. For

Table 2. Retrieval performance on **TREC datasets**, averaged over the 12 topics. Bold-faced is the best run in each row.

\Query type Metrics	Keyword (short)	Keyword (long)	CNF (initial)	CNF (final)
MAP@1000	0.1635[l]	0.1038	0.1815[l]	**0.2017**[l]
P@5	0.5833[l]	0.3167	0.5333[l]	**0.6000**[l]
P@10	0.5333[l]	0.3000	0.5250[l]	**0.5500**[ln]

[l] means the run is significantly better than the long keyword baseline by a two tailed t-test at $p < 0.05$.
[n] means significantly better than long keyword by sign test at $p < 0.05$.

Table 3. Per topic retrieval performance of final Boolean CNF query vs. short keyword query on **TREC datasets**. Bold faced is the better result of keyword and CNF queries in the row.

TREC Topic No.	Keyword (short)		CNF (final) (vs. short keyword)			
	MAP	P@5	MAP	change	P@5	change
352	0.0462	**0.8**	**0.1175**	154.3%	0.6	-25.00%
354	**0.0542**	**0.4**	0.0328	-39.48%	0.0	-100.0%
704	0.2167	0.4	**0.3928**	81.26%	**0.8**	100.0%
751	0.1746	1.0	**0.2170**	24.28%	1.0	0.000%
752	**0.2237**	**1.0**	0.1574	-29.64%	0.8	-20.00%
753	0.3472	0.6	**0.4736**	36.41%	**1.0**	66.67%
758	0.3144	1.0	**0.3187**	1.368%	1.0	0.000%
760	**0.1609**	0.8	0.1279	-20.51%	**1.0**	25.00%
764	**0.1999**	**0.6**	0.0180	-91.00%	0.4	-33.33%
769	0.0143	0.0	**0.4588**	3108%	**0.6**	+inf
799	**0.1850**	**0.4**	0.0946	-48.87%	0.0	-100.0%
805	**0.0247**	0.0	0.0108	-56.28%	0.0	0.000%

Table 4. Retrieval performance with **search engine evaluation**, averaged over the 12 topics. Bold-faced are the better run(s) in each row.

\Query type \ & Metrics Search engine	Keyword (short)		CNF (final)	
	MAP@5	P@5	MAP@5	P@5
Google	0.1586	0.6333	**0.2247**[sn]	**0.8500**[sn]
Yahoo	0.1701	**0.7167**	**0.2041**	**0.7167**

[s] means significantly better than the short keyword baseline by a two tailed t-test at $p < 0.004$.
[n] means also significant by sign test at $p < 0.004$.

Table 5. Per topic retrieval performance in MAP@5 for final Boolean query vs. short keyword query on **Google and Yahoo**. Bold faced is the better result of keyword and CNF queries.

TREC Topic No.	Keyword (short)		CNF (final)			
	Google	Yahoo	Google	change	Yahoo	change
352	0.1133	**0.1300**	**0.2000**	76.52%	0.0000	-100.0%
354	0.1462	**0.1538**	**0.1923**	31.53%	0.0192	-87.52%
704	0.3800	0.3800	**0.5000**	31.58%	**0.4000**	5.263%
751	0.0467	0.1600	**0.2000**	328.3%	**0.2000**	25.00%
752	0.1368	0.1693	**0.1693**	23.76%	**0.2000**	18.13%
753	0.0883	0.0800	**0.2500**	183.1%	**0.1900**	137.5%
758	0.2083	0.2083	0.2083	0.000%	0.2083	0.000%
760	0.2923	0.2923	**0.3846**	31.58%	**0.3077**	5.269%
764	0.0000	0.0200	**0.0833**	+inf	**0.2750**	1275%
769	0.0000	0.0464	**0.0179**	+inf	**0.1940**	318.1%
799	0.1786	**0.1786**	0.1786	0.000%	0.1429	-19.99%
805	0.3125	0.2219	0.3125	0.000%	**0.3125**	40.83%

topic 799 on Yahoo, the CNF query is only slightly worse, returning the only irrelevant result at rank 5.)

The user created Boolean queries outperform short keyword queries consistently, but different Boolean queries improve over the keyword queries for different reasons. The synonym expansion queries are better than keyword because they can solve the mismatch problems of the individual query terms in the keyword query. Topics 354, 751 and 758 are such examples. The restrictive type of Boolean queries outperforms keyword queries because the short keyword queries may match many false positives on the Web. A slightly more restrictive query can remove these false positives while still match enough relevant documents to fill up the top ranks. Topics 704, 753 and 769 are examples.

4.2.4 Discussion

When comparing CNF queries with short keyword queries, on TREC datasets, the difference is not very significant, however, on the search engines, CNF queries are consistently better.

This difference is likely because of two reasons. Firstly, when the users created the CNF queries, they tuned the queries by observing their retrieval results returned from the search engines. Thus, as long as the user makes a serious effort, the tuned Boolean queries will be better performing than the keyword queries on the search engines that the users tuned their queries on. Secondly, only top rank performance was observed and measured with the search engines. Since results deeper down the rank list were not available to the users, they would tend to create highly restrictive queries that improve top precision. This could explain why many of the Boolean queries were more restrictive versions of the short keyword queries. These restrictive queries would likely increase top precision on the search engines (which searched against very large corpora), but would likely decrease lower rank performance as suggested by the deeper evaluations on the TREC datasets. On the much smaller TREC corpora, these restrictive queries will match much fewer documents, thus could even hurt top precision. Overall, the more restrictive Boolean queries perform unstably at both top rank and lower rank levels on the smaller TREC datasets.

This suggests that even though it may seem to the user that a restrictive query would be better, more often than not, synonym expansion is the more robust strategy of query formulation.

5. CONCLUSIONS

Boolean CNF expansion queries have the potential to significantly outperform keyword queries, leading to much more effective retrieval. This paper investigates whether ordinary search users with limited knowledge of CNF queries can formulate effective CNF queries using the WikiQuery interface.

Evaluations on TREC datasets show that versus lengthening the short keyword queries by adding more keywords, creating a Boolean structured query can be significantly more effective at both top and deeper level retrieval accuracy. These Boolean queries are also better performing than the short keyword queries on average. However this difference is not statistically significant on the TREC datasets. Evaluations of the user created Boolean queries

on commercial Web search engines show that these highly precise Boolean queries can consistently and significantly outperform the original short keyword queries in top precision.

Both expansion and restriction query modifications were common when the users created the Boolean queries from the short keyword queries. Some of the expansion queries included many synonyms for each original query term, just like those created by experts [14]. These carefully expanded CNF queries have been shown to outperform keyword queries in precision at all recall levels, because CNF expansion can effectively solve term mismatch, a common problem in retrieval with a large potential [14]. However, even with instructions and the guidance from WikiQuery, users still tended to create less well expanded queries. Users also tended to restrict the original keyword query by introducing phrases or more conjuncts, causing more mismatches between the query and the relevant documents. These restrictive queries might improve top precision, but deeper level evaluation on TREC datasets showed that these restrictive queries do not result in stable improvements at lower rank levels. This tendency to create the less effective restrictive-queries is perhaps one of the reasons why novice users have difficulty creating effective Boolean queries or structured queries.

Use in Text Retrieval

Our results suggest that to improve users' interactions with the search engine, and to facilitate them in creating effective queries, users need to be carefully guided to create CNF expansion queries, and to be explicitly warned against the risky restrictive queries.

Classroom Use

This work used WikiQuery as an educational tool for students with limited knowledge about Boolean queries to learn to create effective Boolean queries in a short time. We observe that most students spent about 40 minutes per topic, trying out new queries and interacting with the search results to find effective formulations. Trial and error using the interactive interface of WikiQuery helped the students quickly and effectively learn the subject.

Open source search tools like WikiQuery and IR education can be mutually beneficial. These tools may become the appropriate playground for educational uses, while classroom uses can also provide a steady stream of traffic for these search tools.

Future Work

The WikiQuery website is still in its early stage. This work as a pilot study can be used to guide and prioritize the development of many new and helpful features for WikiQuery.

Search result presentation needs to be improved to help users quickly grasp why a particular document is returned and what terms in each conjunct of the CNF query are present in the document. Such understandings will allow users to efficiently identify further refinements of the CNF query to improve the results. On a commercial search engine, such user interface changes would be deemed too risky. A research oriented search engine might be the best place to lead the effort.

To further facilitate users in their CNF query creation, synonyms or other related words of each conjunct could be automatically suggested to the user, so that the user only needs to select the highly precise expansion terms out of the suggestions.

To better facilitate users in query refinement, novel interfaces that can automatically extract or highlight candidate expansion terms in result snippets or documents can be useful.

To help users decide whether to include one particular expansion term into a conjunct or not, tools that can compare rank list changes before and after a query change will be useful. In particular, tools that can present deeper rank level changes will enable the user to more accurately gauge the overall retrieval accuracy.

The WikiQuery website may also allow users to subscribe to the result pages of each CNF query, so that whenever a new relevant page appears on the Web, the user will be notified. This is the equivalent of a traditional routing task, and can be easy implemented given that search engines like Google already support user subscription to search engine result pages.

6. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, 1999.

[2] J. R. Baron, D. D. Lewis and D. W. Oard. TREC 2006 Legal Track Overview. In *Proceedings of the fifteenth Text REtrieval Conference (TREC '06)*, 2007.

[3] D. C. Blair and M. E. Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of ACM*, 28(3): 289-299. 1985.

[4] C. L. A. Clarke, G. V. Cormack and F. J. Burkowski. Shortest Substring Ranking (MultiText Experiments for TREC-4). In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*. 1996.

[5] H. A. Feild, J. Allan and R. Jones. Predicting searcher frustration. In *Proceedings of 33rd annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '10)*. 34-41, 2010.

[6] S. Harter. *Online Information Retrieval: Concepts, Principles, and Techniques*. Academic Press. San Diego, California. 1986.

[7] M. Hearst. Improving full-text precision on short queries using simple constraints. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR '96)*, 1996.

[8] M. Hearst. *Search User Interfaces*. Cambridge University Press. 2009.

[9] W. Lancaster. *Information Retrieval Systems: Characteristics, Testing and Evaluation*. Wiley. New York, New York, USA. 1968.

[10] C. D. Manning, P. Raghavan and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press. 2008.

[11] M. Mitra, A. Singhal and C. Buckley. Improving automatic query expansion. In *Proceedings of 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '98)*. 206-214, 1998.

[12] P. Reisner. Construction of a growing thesaurus by conversational interaction in a man-machine system. *Proceedings of the American Documentation Institute*, 26th annual meeting, Chicago, Illinois, 1963.

[13] L. Zhao and J. Callan. Term necessity prediction. In Proceedings of the *19th ACM Conference on Information and Knowledge Management*. 2010.

[14] L. Zhao and J. Callan. Automatic term mismatch diagnosis for selective query expansion. To appear *Proceedings of 35th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '12)*. 2012.