ABSTRACT
In this article, we report on the study of Terrier search engine for Quranic verse retrieval problem. However, the results produced by Terrier search engine are not yet completely satisfying; an extensible framework has been proposed that incorporates feedback from the user to generate a better query. In this article, we integrate the feedback framework with the Terrier search engine and extend the existing browser-based interface of Terrier search engine to support explicit relevance feedback, re-evaluate the query when new feedback is available. We also modify the existing Terrier search engine to support queries; the evaluation of the query should reuse the partial results from the evaluation of the original query.

KEYWORDS—Relevance Feedback, Web Based User Interface, Information Retrieval, Terrier.

1 INTRODUCTION
The need to access information anywhere at any time is very important today in order to facilitate effective decision-making. The areas such as Islamic environment also require adequate information, in which such system usually relies on the expert to provide the answer to question. In this article, it is argued that the adoption of a collaborative relevance feedback algorithm into an Islamic question answering (QA) system could approximate the Q&A services provided previously by the experts. In collaborative relevance feedback, a collection of relevance judgments to a question provided by normal users (non-expert) can be used to train the QA system. Search engines employing standard information retrieval techniques can help to retrieve some particular piece of information quickly. However, the common phenomenon is that users find it difficult to express their actual information need as a query. The field of Information Retrieval (IR) addresses the general problem of how to retrieve information, which is relevant to a user need, from a given repository of information, such as a document collection [1].

A web based user interface is designed to facilitate effective user interface for information retrieval; the interface was designed in such a way that the user can search for the information needed. The issue of relevance and validation of information is sensitive in Islamic context. It must be based on the source of sharia, which are Quran, its tafsir, the collection of Hadiths, the Fatwa collections, the Websites dedicated to Islamic issues and also the collection of Blogs by the experts.

The rapid expansion of the Web has led to an increase in the number of casual searchers that are using Web search engines. “Many searchers are inexperienced and can have difficulty expressing their information need”, and Jansen et al. [2] showed that 62% of queries submitted to the Excite web search engine contained only one or two terms. This kind of queries lack useful search terms that may lead to ineffective search [13].

In practice, relevance feedback can be very effective but it normally relies on users assessing the relevance of documents and indicating to the system that documents contains the relevant information. In the real-life, the users may be unwilling to browse to web pages to gauge their relevance. Such task imposes increased burden and also increased cognitive load [9].

It is believed that there are some issues or questions that can be answered by other trusted
users. For instance, a simple question about ablution (wudu’) can be answered by a trusted colleague without the need for the expert. This research attempts to provide the solution for collaborative Islamic QA system, which takes into account the opinions or advises from normal users.

The scope of this paper is to investigate the effectiveness of state of the art IR techniques for Islamic collaborative information retrieval problem. The test collections were built based on manually indexed topics of Quran. In addition, a few standard performance measures used in IR researches are used and discussed in the context of the Quranic verse retrieval problem. Furthermore, the Terrier [3], a state of the art IR system developed by the University of Glasgow, is used in the experiment to discover the effectiveness of the verse retrieval system.

The rest of this article is organized as follows, Section 2 presents summary of the related work. Section 3 introduces the design and methodology used. Section 4 analyzes the results from the experiments. Finally, Section 5 summarizes the paper along with the future research goals.

2 RELATED STUDIES

In this section we provide the background information for the different concepts and techniques used and discussed through the next chapters of this article. We start with an introduction to the Web information retrieval. We illustrate the process of relevance feedback and how it contributes to enhance the retrieval accuracy of search engines. We also show how the relevance feedback process could be used in the web search engines

2.1 Web Information Retrieval

HTML (Hypertext Mark-up Language) is a mark-up language for documents containing structured information. This Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays [5]. Almost all the documents have some structure. A mark-up language is a mechanism that identifies structures in a document. The HTML specification defines a standard way to add mark-up to documents [15]. HTML uses a Document Type Definition (DTD) or an XML Schema to describe the data. HTML with a DTD or HTML Schema is designed to be self-descriptive. The following shows an example HTML document [15] in [6]:

```
The first line in the document - the HTML declaration - defines the HTML version and the character encoding used in document. The document In this case conforms to the 1.0 specification of HTML and uses the ISO-8859-1 (Latin-1/West European) of the character set. The next line also describes the root element of the document (like it was saying: this document is a note”) [6]. The next four lines describe four child elements of the root (to, from, heading, and body). And the last line defines the end of the root of element finally. HTML has recently seen increasing importance to represent large amounts of semi-structured and textual information in scientific data repositories, Digital Libraries and across the Web. As a consequence to that, the research on XML information retrieval has been increasing and has become popular in recent years. Many people working on this research area have developed retrieval tools and search engines that employ the ranked-retrieval paradigm used in text-based information retrieval search engines rather than simply using the Boolean retrieval as in XPath and XQuery, examples of such systems are TopX [7], FleXPath [9] and XRANK [11] in [3].
```

Terrier search engine is an example of a search engine for ranked retrieval. Terrier supports both document-level and element-level of retrieval. If we submitted a query using Terrier and we used the document retrieval mode, So, following the discussion before; the top-10 results would be the top-10 ranked documents containing elements that have content relevant to our query. Also, we can submit the same query but in the element such that the retrieval mode of the top-10 results would be the top-10 ranked elements containing content relevant to our query.

The Terrier [11] [12] is a state of the art search engine available as an open source from
Glasgow University. It is a probabilistic based retrieval model with a number of sophisticated Divergences from Randomness (DFR) weighting model. It is based on the idea that the more divergence within document term-frequency (TF) from its frequency within the collection, the more information carried by word in the document [13]. The search engine has demonstrated impressive performance in many retrieval evaluation forums such as Text Retrieval Conference (TREC) and Cross Language Evaluation Forum (CLEF).

### 2.2 Relevance Feedback

Relevance feedback is a technique that typically either modify a query automatically, [7], or present the user with a set of new terms to allow the user to interactively modify their query [10]. Relevance feedback is the process in which users evaluate (mark or select) a small set of search results as relevant or non-relevant with respect to their informational need. Automatic relevance feedback techniques can suffer from the fact that users are not willing to use these techniques. In particular this can be because the user does not understand the relation between the relevance assessment and the effect of relevance feedback [4]. Interactive query modification has often been preferred on the grounds that it allows the user more control over their search, [4]. However, searchers often do not fully utilize interactive query modification techniques; either because they do not know how to choose good new query terms [1], this is because they do not understand the effect of the new terms will have on their search [7].

We can differentiate between three types of relevance feedback: explicit feedback, implicit feedback, and pseudo (blind) feedback [3].

**Explicit Feedback**

In explicit feedback, the user explicitly marks (some) search results and indicate as relevant or not relevant. Many empirical evaluations have shown that explicit feedback is an effective way to improve the retrieval accuracy as it uses strong evidence from the user about what is relevant and what is not relevant [13].

**Implicit Feedback**

In this case, the system tries to automatically infer what is relevant and what is not relevant from the user's interactions with the search results. This mainly relies on heuristics, which is assuming that search results that the user clicks are relevant and those non clicked results that were ranked higher than a clicked result are non-relevant [3]. We can see that implicit feedback neither asks for the user's explicit relevance judgment nor categorically assumes that top ranked documents of baseline retrieval are relevant. Implicit feedback intelligently infers instead, the user's information need through those user's interactions with the search results. However, there is a caveat for implicit feedback. We need to carefully analyze those hints and do not incorporate noise into the new query that may even hurt the retrieval performance.

**Pseudo Feedback**

Pseudo feedback simply assumes some top ranked results to be relevant and thus does not require the user to explicitly mark the search results. A lot of empirical evaluations show that pseudo feedback generally, can outperform the baseline retrieval but not always. Pseudo feedback however, is not as effective as explicit feedback [2].

### 3 DESIGN AND METHODOLOGY

The Terrier [11,12] is a state of the art search engine available as an open source from Glasgow University. It is a probabilistic based retrieval model with a number of sophisticated Divergences from Randomness (DFR) weighting model. It is based on the idea that the more divergence within document term-frequency (TF) from its frequency within the collection, the more information carried by word in the document [13].

#### 3.1 Effective User Interface

The feedback engine was implemented in Java (JSP) and the Terrier search engine. The implementation of the feedback engine requires important information about the elements to be pre-computed which is the unique identifier for the element and its document. The data are stored in a database table with schema (quran text) that contains one tuple for each distinct
term of an element. On the database side, indexes on (chapter) are provided to efficiently find (English) for an element sura and on (verse) to efficiently collect all elements of a document. For each element with known relevance, the implementation of the feedback search engine will first loads all the complete content of the element's document and computes all possible expansions in memory with the English for the candidate loaded from the database. We cache the English and the content of elements once they get loaded from the database in order to avoid loading them again in the new feedback cycles.

3.2 Web Based User Interface
The Web-based interface of the Terrier search engine that we extend in this article is implemented using the Terrier 3.0. Terrier is an open source search engine written by using Java development framework which makes writing JSP based application interface easier. The Java compiler translates the Java application written in Terrier to browser-compliant JavaScript and HTML. We use a Java class coined Terrier Feedback Result to represent the elements with known relevance gathered from the user through the interface of Terrier search engine. This class contains fields to store data necessary for our feedback engine.

3.3 Using Session Context
As stated earlier we initialize the evaluation of the expanded feedback query from the evaluation of the initial keyword query by storing the state of Terrier (i.e. index lists, index cache, candidates and the top-k items) from the execution of the initial query. We store the state data using session variables on the server side. We use a session variable for storing the index lists, and another one for storing the top-k items. We use this data to initialize the new corresponding variables from new execution, which also run on the server side. The search engine was connected to MySQL database having three tables; each contains the Quranic translation data for each languages, Arabic, English, Malay and Hausa language.

4 RESULTS AND DISCUSSION
In this section we discuss the detail of the experiments, using the Terrier IR system configured based on its default settings. For the Malay translated Quran, the Terrier is configured to remove Malay stop words and to set the parameter to its default value, 1.0. The collection is not stemmed, as the Malay stemming algorithms have not been adopted into the system.

For the English translated Quran, the system is configured to remove stop words and also to apply stemming as provided by the Terrier by default. In addition, the parameter is set to its default value. Therefore, there will be three IR system being evaluated; Malay, English and Hausa. The evaluation of feedback engine were conducted on a 4.0 GHz Intel Window 7 PC, with 4 GB of memory. Index lists were stored in MySQL server running on a different PC. We used the database collection of the Holy Quran in MySQL and XML format. We chose some queries from the manually indexed topics used in [16].

4.2 Evaluation Methodology and Results
We evaluated each query with the Terrier search engine, which is integrated in Web-based interface. Relevance feedback information was provided for the initial results by marking some results as relevant or non-relevant. The generated feedback queries were evaluated two times: one time with the original version of the Terrier search engine (without our modification) and the other time with the improved Terrier version that supports relevance feedback. Table 1 and 2 show some results of our evaluations from the top-k results by selecting the best-10 candidates from expansion of each query.

Based on the display result we compare and judge some relevance from the collected data to that of researcher, the following result were obtained. The result displays in Arabic, English, Malay as well as Hausa (Nigeria language) once the user key in the query. Precision and recall were use to evaluate the performance of our system. Precision and recall are the most common measurements to evaluate the
effectiveness of IR systems. According to [15], relevancy of retrieved verses will be assumed to have its broader meaning of 'aboutness' and 'appropriateness'.

\[
\text{Precision} = \frac{\text{# of relevant document retrieved}}{\text{# of document retrieved}}
\]

\[
\text{Recall} = \frac{\text{# of relevant document retrieved}}{\text{# of relevant document}}
\]

We use the first thirty result generated in our case study, after which we compute the precision and recall. The results of the precision and recall were showed in the Table 1 and 2.

**TABLE 1 : Initial precision and recall**

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
<td>0.33</td>
</tr>
</tbody>
</table>

As depicted in Table 1, the data from [16] obtained based on the relevance judgment when compared to that of researcher it’s seem in the first top thirty result only one result match. The rest matched were found in the next top forty, fifty and soon. The judgment was made based on the researcher persepctive, which may or may not be correct, since the research is not an expert in the case.

**TABLE 2 : Precision and recall after feedback**

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.83</td>
<td>0.30</td>
</tr>
<tr>
<td>0.43</td>
<td>0.30</td>
</tr>
<tr>
<td>0.86</td>
<td>0.30</td>
</tr>
<tr>
<td>0.80</td>
<td>0.30</td>
</tr>
<tr>
<td>0.50</td>
<td>0.30</td>
</tr>
<tr>
<td>0.60</td>
<td>0.30</td>
</tr>
<tr>
<td>0.56</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Based on Table 2, the precision has been improved dramatically while recall remains the same for all queries. We believe that the reason behind this improvement was due to the fact that the number of relevant verses would be increased after applying the relevant feedback.

**TABLE 3 : Comparison of the precision before and after**

<table>
<thead>
<tr>
<th>Query</th>
<th>Precision Before</th>
<th>Precision After</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>0.33</td>
<td>0.83</td>
</tr>
<tr>
<td>B2</td>
<td>0.33</td>
<td>0.43</td>
</tr>
<tr>
<td>A3</td>
<td>0.33</td>
<td>0.86</td>
</tr>
<tr>
<td>G4</td>
<td>0.33</td>
<td>0.80</td>
</tr>
<tr>
<td>F5</td>
<td>0.33</td>
<td>0.50</td>
</tr>
<tr>
<td>B6</td>
<td>0.33</td>
<td>0.60</td>
</tr>
<tr>
<td>P7</td>
<td>0.33</td>
<td>0.56</td>
</tr>
</tbody>
</table>

In the case of recall, there was no change in the values since we conduct our experiment only on the top thirty retrieved results. This improvement may not yet be concluded to be good enough since the data here is small. But with the different we got here signifies that a better performance can be obtained if generalized.

**Figure 1 Precision comparison**

Figure 1, shows the difference between the precision before applying the relevance judgments and how it had been changed after using the relevance feedback.

5 **CONCLUSION AND FUTURE WORK**

We presented our approach for integrating relevance feedback framework with the Terrier search engine. Our approach makes use of the explicit feedback present from the user's judgments for the search results, through the Web-based interface of Terrier search engine, in
order to improve search results ranking. We do pre-computations necessary to initialize the feedback engine. The feedback engine uses the elements with known relevance to construct the feedback query.

We exploit partial results from the evaluation of the initial queries using the Terrier search engine to evaluate the feedback queries. In future, we are going to extend the framework for relevance feedback in order to support queries with content and structural constraints. Currently, the feedback framework that was used in this article was limited only to queries based on Quranic information. In the multiple feedback cycles, we used the same initial keyword-based query to generate the refined queries based on the feedback. Furthermore, we can extend this to include Hadith, Fatwa, and other Islamic information in order to facilitate effectiveness of relevance feedback.

Another potential work includes applying to other languages rather than Malay and English. As well, we plan to use experts’ judgments from Islamic scholars and apply for complete automatic query answering systems.

REFERENCES


[3] Osama Sammodi Incremental Relevance Feedback for TopX thesis submitted to Saarland University, Department of Computer Science Max Planck Institute for Computer Science, Saarbrücken, Germany Databases and Information Systems Group (AG5)


