The Usage of Formal Methods in Quran Search System

Alaa Al-Gharaibeh, Ahmad Al-Taani, Izzat Alsmadi

Faculty of Information Technology, Yarmouk University, Jordan.

{alaa_mg@yahoo.com, ahmadta@yu.edu.jo, ialsmadi@yu.edu.jo}

Abstract

Formal methods are mathematically-based techniques for the description and verification of the specification of software and hardware systems. There are many research papers discuss the usage of model checkers in critical systems. This study shows how to use the formal methods for Natural Language Processing in a Quranic search system (QSS). The Z notation is used for expressing the formal specifications of the three search techniques text-based, stem-based, and synonyms-based systems which are used in a QSS. QSS allows the user to search about keywords in the holy Quran and retrieve the relevant verses. Z/EVES tool is used for checking, and analyzing Z specifications.

Keywords: Formal Methods, Natural Language Processing, Software Engineering, Information Retrieval.

1. Introduction

Formal methods allow a software engineer to create a specification that is more complete, consistent, and unambiguous than those produced using conventional methods. Set theory and logic notation are used to create a clear statement of requirement. This mathematical specification can be analyzed to improve correctness and consistency [1].

The Z notation is a language and a style for expressing formal specifications of computing systems based on a typed set theory, and its key features is the notion of a schema. A schema consists of a collection of named objects with a relationship specified by some axioms, and Z provides notations for defining schemas and later combining them in various ways [12].

A QSS allows users to search about keywords in the holy Quran. Figure 1 shows the architecture of this system. First, the holy Quran is analyzed its keywords are represented in a database file. Second, the light stemming algorithm is used to find the stems of the words. Finally the user query is analyzed. The search process is performed using three approaches; text-based, stem-based, and synonyms-based approaches. The text-based system approach is based on full words, the synonyms-based approach is based on the synonyms of the words, and the stem-based approach is based on the stem of the words. The light stemming algorithm is used to find the stem of the word.

The formal method is used to create a specification of the QSS. The Z notation is used for expressing the formal specifications of the three search approaches text-based, stem-based, and synonyms-based approaches.

We described each approach in schema and describe the operation of each type informally, then define the syntax of the operations in the interface and their parameters, then define the operation semantics by defining axioms which characterize behavior.

Z/EVES tool is used for composing and checking Z specifications of the QSS. Each type of specifications entered and checked one paragraph at a time.

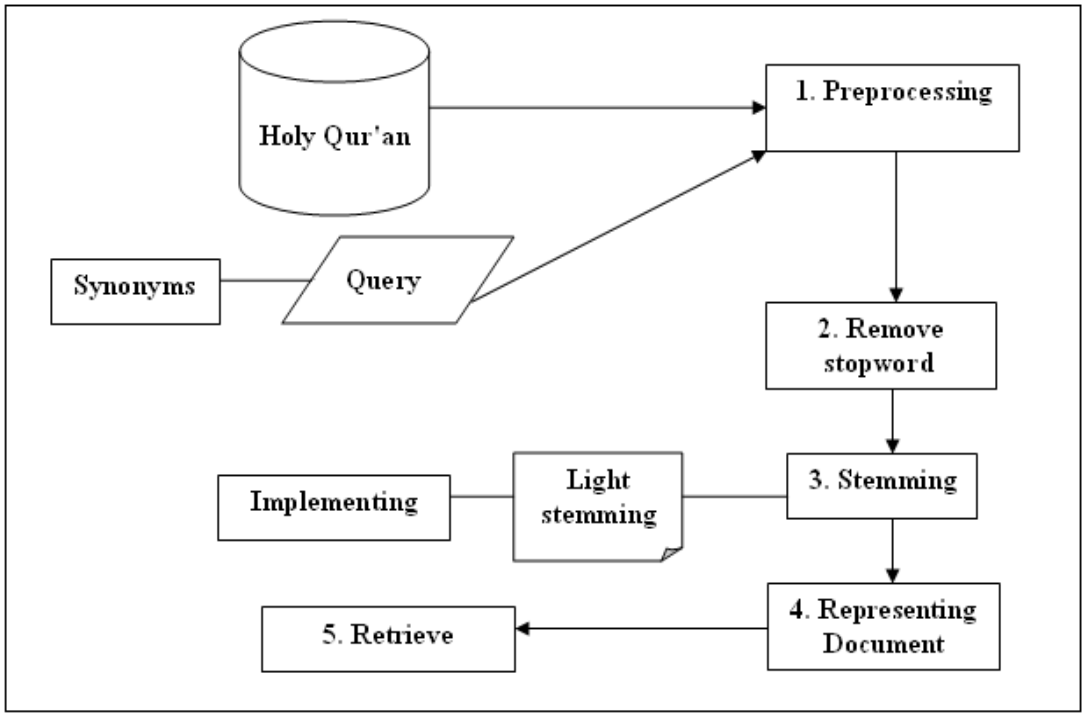

Figure 1: Quran Search System [13]

2. Literature Review

El Qadi *et al.* [2] described a mechanism to improve Information Retrieval (IR) on the web. The method is based on Formal Concepts Analysis (FCA) that makes semantic relations during the queries, and allows a reorganizing in the shape of a lattice of concepts, the answers provided by a search engine. The authors proposed an algorithm that allows a formal clustering of the data sources, and the results are classified by order of relevance. The control of relevance is exploited in clustering.

Poshyvanyk [3] presented an approach which combines Formal Concept Analysis (FCA) and Latent Semantic Indexing (LSI) to address the problem of concept location in source code. In the proposed approach, LSI is used to map the concepts expressed in queries written by the programmer to relevant parts of the source code, presented as a ranked list of search results. Given the ranked list of source code elements, their approach selects most relevant attributes from these documents and organizes the results in a concept lattice, generated via FCA.

Quintana *et al.* [4] presented a framework for modeling the cognitive search tasks involved in hypertext searching. The authors proposed a model for the user, the document, and the interface. The information that needs to be contained in these models has been defined and they described the interaction of these models. A predictive model has been presented that allows for estimating retrieval times from hypertext documents based on a model of the document, user expertise, and user interface.

Lavrenko *et al.* [5] discussed a model of retrieval that bridges a gap between the classical probabilistic models of information retrieval, and the emerging language modeling approaches. The authors explained why classical models with their explicit notion of relevance may potentially be more attractive than models that limit queries to being a sample of text.

Miller *et al.* [6] presented a new method for information retrieval using hidden Markov models (HMMs). The authors developed a general framework for incorporating multiple word generation mechanisms within the same model. Then they demonstrated that an extremely simple realization of this model substantially outperforms standard tf : idf ranking on both the TREC-6 and TREC-7 ad hoc retrieval tasks.

Ponte *et al.* [7] proposed an approach to retrieval based on probabilistic language modeling. The authors estimated models for each document individually. Their approach to modeling is non-parametric and integrates document indexing and document retrieval into a single model. The advantage of this approach is that collection statistics which are used heuristically in many other retrieval models are an integral part of their model.

Fuhr *et al*. [8] presented the new query language XIRQL which integrates all IR-related features, and described the concepts that are necessary in order to arrive at a consistent model for XML retrieval. For processing XIRQL queries, the authors described a path which serves as a starting point for query optimization. In parallel, XIRQL can be extended to include the data-centric features from XQuery.

Jianyun Nie [9] described the general functionalities of information retrieval systems by construct a more general model. The implicit basis in all information retrieval systems is considered as logical implications. The measurement of the correspondence between a query and a document thus becomes the estimation of the implication strength. This estimation can be suitably described in terms of modal logic. The model is shown to be more general than the existing ones, such as the Boolean model, the probabilistic model.

Wong *et al*. [10] proposed a generalization of the VSM, called the GVSM (general vector space model). The developments provide a solution for the computation of a measure of similarity between terms and for the incorporation of these similarities into the retrieval process. The major strength of the GVSM derives from the fact that it is theoretically sound and elegant.

Noordin *et al*. [11] proposed a system design for retrieving Quranic texts and any knowledge that derived or cites al-Quran. The objectives were to survey the Websites offering access to Quranic texts on their structure and linkages, and to propose a system design for retrieving Quranic texts.

3. The specifications of the QSS

Z notation is used for expressing the specifications of three search techniques; stem-based system, synonyms-based system and text-based approaches that are used in QSS. The processes of the QSS are described as follows [13]:

1. Select the text file that contains Qur'anic text as search data.

2. Build the stop word table.

3. Build the index table

3.1. Read the text file verse by verse.

3.2. Read the verse word by word.

3.2.1 IF the word is not an Arabic word THEN consider this word as a useless word.

3.2.2 IF the word contains digits THEN consider this word as a useless word.

3.2.3 IF the word length is less than three characters THEN consider this word as a Useless word

3.2.4 Remove diacritics.

3.2.5 Normalize the word.

3.3 Apply the stop word test to check if this term is a stop word, if the term is a stop word, discard it. Otherwise, if you are using full word indexing go to step 3.4 In case of root indexing go to step 3.3.1.

3.3.1 Remove prefixes and recursively remove suffixes, then go to step 3.4.

3.4 Gather the words of each verse

3.5 Insert the group of words of each verse to the index table.

3.6 Match the word of the query with the words in the index table.

3.7 Retrieve the verses which may more relevant to the user demand.

3.1 Stem-based system specification

In the stem-based system, the search method is based on the root of the words, each word of the user query and document is preprocessed (removing stop words, stemming). Each root word of the user query is matched to the root word in the index table then the verses that have the same root word are retrieved.

The specification of stem-based system is expressed by using Z schema as shown below. The name of schema is STEM. There are many operations used in the QSS to preprocess the document and query:

Create: create the index table which contains the verses and verses after preprocessing.

Input: the user input the word in the query.

Isarabicword: the system makes sure that the word is an Arabic word.

Isuselessword: the system considers any word contains digits or the word length less than 3 letters as an article and thus a not important word.

*Remove diacritics**:** remove the diacritics from every word in the document**.***

Remove-stopword: For every word in the text if it is a stop word then the word is removed

*Remove prefixes**:** removes a set of prefixes. After removing these prefixes, the system checks if the word length is less than 3 letters, in this case these prefixes considered as a main part of the word and so the removed prefix returned back to the word.*

*Removes suffixes**:** remove a set of suffixes from the tail of word. The system checks if the word length is greater than 3 letters in order to prevent remove a main part of the word.*

Retrieve: retrieves the verses which contain the root word that is matched with the root words of the user query.

3.2. Synonyms-based system specification

In synonyms-based system, the search method is based on the synonyms of the word. MS WORD Arabic thesaurus is used to find the synonyms for each word in the user query. Each synonyms word in the user query is matched to the same word in the index table then the system retrieves the verses which have the same word.

The specification of synonyms-based system is expressed by using the Z schema as shown below. The name of schema is SYNONYMS. There are many operations are used, some of them are the same with the operations in stem based system. The operations in synonyms specification those are different from stem specification:

Findsynonyms: find the synonyms of the word query from Arabic thesaurus.

Match: match the synonyms of the word with the word in index table.

Retrieve: retrieve the verses which have the word that match with the synonyms of the word query.

3.3. Text-based system specification

In the text-based system, the search method is based on the full word, each word in the user query is matched to the same word in the index table then the system retrieves the verses that have the same word. The specification of the text-based system is expressed by using Z schema. The name of schema is EXACT MATCH. The difference between this system and the previous systems is the operation of exact match which match the exact word in user query with the word in the index table.

4. The verification of the QSS using Z/ EVES

Z/EVES tool is used for composing and checking Z specifications of the QSS. The specifications are entered and checked one paragraph at a time.

Each Z schema has A name, A declaration part, and a A predicate part.

First, we define the type Word and Root as a free type definition then check them (see Figure 2).

Second, we write *Indextable* schema to define a database; in declaration part, we define a table as a set of pairs. Each pair represents a word followed by its root, then check them (see Figure 3).

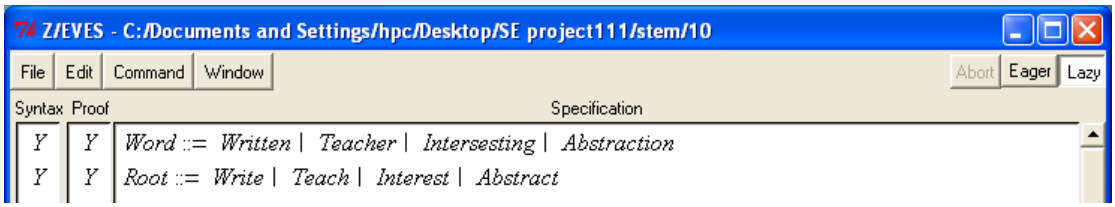

Figure 2: Specification types

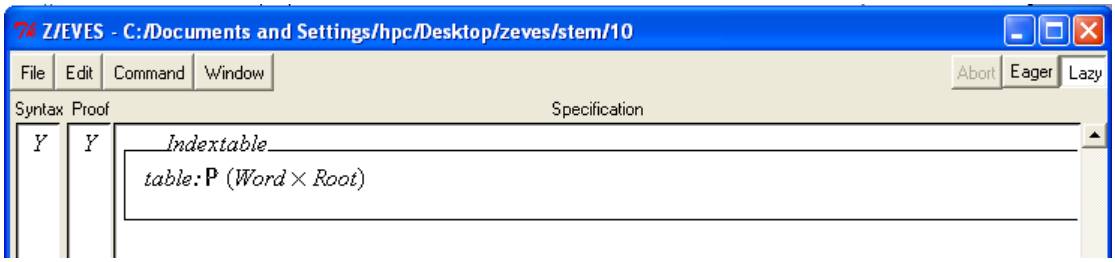

Figure 3: Checking the database

Third, we write the *Addtotable* schema to add data to a database. In declaration part we use ΔIndextable for both uses and changes the set *table* from above. In predicate part the (+) operator (‘plus in a circle’) overwrites the existing pair using word?.

The *Stem* schema is written to specify how to retrieve data from the table. In declaration part, query? is an input of type Word, and out! is an output of type Root. In predicate part, dom is the domain of the relation *table*: the set of all data items appearing on the left hand side of the relation (see Figure 4).

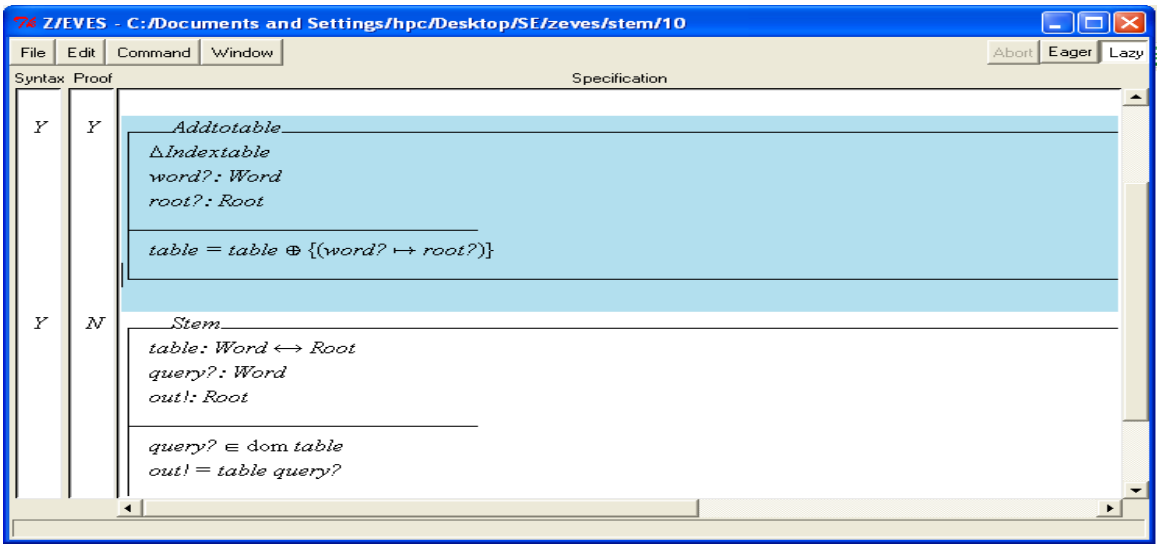

Figure 4: Retrieving data

5. Conclusion

The formal method is used to create a specification of the QSS. The Z notation is used for expressing the formal specifications of the three search approaches; text-based, stem-based, and synonyms-based used in the QSS. Z/EVES tool is used for composing and checking the Z specifications of the QSS. Z/EVES doesn't support Arabic.

6. References

[1] Roger Pressman, Software engineering: a practitioner's approach, McGraw-Hill, 2009.

[2] El Qadi A., Aboutajdine D. and Ennouary Y., "Formal Concept Analysis for Information Retrieval", International Journal of Computer Science and Information Security, 7(2): 119-125, 2010.

[3] Poshyvanyk D. and Marcus A., "Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code", Proceedings of the 15th IEEE International Conference on Program Comprehension (ICPC '07), pp.37-482007.

[4] Quintana Y., Kamel M. and McGeachy R., "Formal Methods for Evaluating Information Retrieval in Hypertext Systems", Proceeding of the 11th annual international conference on systems (SIGDOC'93), pp. 259-272, 1993.

[5] Lavrenko V. and Bruce Croft W., "Relevance-Based Language Models", In proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 120-129, 2001.

[6] Miller D., Leek T. and Schwartz R., "A Hidden Markov Model Information Retrieval System", In proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 214-221, 1999.

[7] Ponte J. and Bruce Croft W., "A Language Modeling Approach to Information Retrieval", In proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275-281, 1998.

[8] Fuhr N. and Großjohann K., " XIRQL: A Query Language for Information Retrieval in XML Documents", In proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 172-180, 2001.

[9] Jianyun N., "An information retrieval model based on modal logic", Information Processing & Management, 25(5): 477-491, 1989.

[10] Wong S., Zlarko W., Raghavan V. and Wong P., "On Modeling of Information Retrieval Concepts in Vector Spaces", ACM Transactions on Database Systems, 12(2): 299-321, 1987.

[11] Noordin M.F. and Othman R.," An Information Retrieval System for Quranic Texts: A Proposed System Design", In proceedings of the 2nd International Conference of Information and Communication Technologies (ICTTA'06), pp.1704 - 1709, 2006.

[12] Spivey J., "Understanding Z: a specification language and its formal semantics", Cambridge University Press, New York, NY, USA, 1988.

[13] Al-Gharaibeh A., "Searching about concept and keywords in the holy Quran", Master graduation project, department of computer science, Yarmouk university, 2010.