

Syntax and Semantics Question Analysis Using User Modelling and Relevance Feedback

Syarilla I. Ahmad Saany[#], Ali Mamat^{*}, Aida Mustapha^{*}, Lilly S. Affendey^{*}, M. Nordin A. Rahman[#]

[#]*Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Tembil, Besut, Terengganu, Malaysia*
E-mail: syarilla@unisza.edu.my, mohdnabd@unisza.edu.my

^{*}*Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia*
E-mail: ali@upm.edu.my, aida@fsktm.upm.edu.my, lilly@upm.edu.my

Abstract— A Question Answering (QA) system aims to provide relevant answers to users' natural language (NL) questions by consulting its knowledge base (KB). Providing users with the most relevant answers to their questions is an issue. Many answers returned are not relevant to the questions, and this issue is due to many factors. One such factor is the ambiguity yield during the semantic analysis of lexical extracted from the user's question. The existing techniques did not consider some of the terms, called modifier terms, in the user's question which are claimed to have a significant impact of returning the correct answer. The objective of this study is to present the syntax and semantic question analysis using user modelling (UM) and relevance feedback (RF). This analysis interprets all the modifier terms in the user's question in order to yield correct answers. A combination of UM and RF is used to increase the accuracy of the returned answer. UM helps the QA system to understand the user's question and manage for question adjustment. RF provides an extended framework for the QA system to avoid or remedy the ambiguity of the user's question. The analysis utilizes Vector Space Model (VSM) to semantically interpret and correctly converts modifier terms into a quantifiable form. The finding of this analysis demonstrates a good precision percentage of 94.7% in returning relevant answers for each NL question.

Keywords— syntax and semantic analysis; question analysis; question answering

I. INTRODUCTION

The emergence of ontology research activities, especially in the Question Answering (QA) systems are progressing towards solving research issues in handling complex questions such as comparative, evaluation, superlative and negation question types [7], [11], [12], [18]. This is not an easy task to automatically capture the semantics lies in a complex question structure. In the context of this research, a question is considered to be simple if the answer is a piece of information that has been located and retrieved directly as it appears in the information source. On the other hand, a question is considered complex if its answer needs more on elaboration [13]. The semantic information contained in the user Natural Language (NL) question may miss or lose during the question analysis process. As in [8], complex questions are analyzed and parsed to multi simple questions which later the existing techniques are used to answer them. Handling complex question may comprise inferences in terminology, analyses the properties or attributes involved before an answer can be drawn [12]. The semantic meaning of this question structure has to be thoroughly analyzed, semantically interpreted and converted into an executable

query so that the potential answers can be obtained from the corpus or knowledge base [12]. In [7], the FREyA (Feedback, Refinement and Extended Vocabulary Aggregation) QA system relies on the user feedback to disambiguate any ambiguity exists in the complex question.

Some of the QA systems perform user modelling to retain individual user information, experiences, common goals and requirement behaviors that characterize the user to a certain style of query statements. User modelling is important because users mostly have little knowledge of the contents and structures of the knowledge base. Besides user modelling (UM), relevance feedback (RF) is another mechanism used to improve the performance of the QA system. RF is used for rating the relevancy of answers with respect to the query [4], [9]. Based on the feedback given by users, the QA system will re-rank the answers and present the results again to the user. RF is useful to users who do not possess prior knowledge of the knowledge base (KB). RF is either applied after the system has produced the results (answers) based on the natural language question submitted by the user or is exploited to interpret the questions.

In QA system, forming the right queries from a given user question is crucial. These queries are responsible for getting

the right set of documents or information which contains potential answers. Most of the former research efforts in QA were concentrated on finding the correct answer. Thus very much attention was paid to the question analysis and processing [6]. Most research agrees that the question analysis and processing component is one of the core engines in the QA systems. In NL interface of the QA, the user's question can be very specific or not quite yet clear either to the user himself or to the QA system.

Currently, the major focus in QA research is advancing towards complex question handlers such as in negation questions [11] or comparative and evaluative questions [7], [12]. An example of negation question is "Are there any AutoCAD jobs open?", whereas an example of a comparative question is "count the states which have elevations lower than what Alabama has?" Processing the comparative question such as the latter example given above or the evaluative question like "could you tell me what is the highest point in the state of Oregon?" involve comparison and evaluation of one or more criteria.

Evaluating comparative and evaluation questions involve inferences in terminologies before the QA system is able to return an answer. The inferencing process includes determining the properties or attributes required for evaluation, computing the associated values of entities or objects and comparing the entities or objects and their values depending on the evaluation of one or more criteria [12]. This means the semantics of comparative and evaluative questions have to be correctly interpreted and converted into a representation based on quantifiable criteria before the answer could be retrieved from the knowledge base (KB).

This scenario has become another main concern in QA. The user may provide question with imprecise representations. This requires expansion, modification or adding information to the question specification. For example, to interpret a question of "How big is Alaska" will depend on either, the context of the question (user's intent) or the structure of the knowledge base. The word big may refer to the size of Alaska or big can also mean the population of Alaska. In this scenario, QA needs to have equipped with complex question handler on how to quantify and evaluate specific term ('big' is a specific term) contained in the user's NL question.

Handling NL question requires semantic analysis, interpretation, and transformation into an executable query so that the potential answers can be obtained from the corpus or knowledge base. The purpose of this paper is to present a syntax and semantic question analysis for QA which utilizes the combination of user modelling and relevance feedback approach. The remainder of this paper is organized as follows Section 2 discusses the related works in QA system. The syntax and semantic question analysis will be presented in Section 3. Section 4 discusses the experiment and results. Section 5 sums up the paper with conclusions and future works.

Many QA systems use an intensive Natural Language Processing (NLP) techniques and information extraction (IE) approach. Since the advent of ontology, recent studies in QA systems have utilized ontology in the development of QA systems. This can be seen in the works of [3], [5], [7], [10], [11], [18] and much more. During the early time, many QA

systems used ontology as a mechanism to support query expansion [18].

During the early time, many QA systems used ontology as a mechanism to support query expansion [14]. In general, ontology is used to provide and share domain knowledge which is also routed through for finding answers to the questions. Instead of failing to provide an answer, an ontology may assist in finding the value or semantic meaning of the term or relation that are looked for. Many researches have shown that ontology promotes the semantic capability of a QA system [7], [10], [18].

AquaLog is the first ontology-based QA system that relies on the knowledge encoded in the underlying ontology and its explicit semantics for question analysis and answer retrieval [18]. In Aqualog, it utilizes linguistic tools (e.g. GATE) and resources (e.g. WordNet) to annotate the terms and relations in an NL question. From the analysis of the AquaLog QA, some user questions such as "Which research areas bring in the most funding?"; "Who are the main researchers in the semantic web research area?"; "What are the new projects in KMi?" and "Who works on the same project as Enrico Motta?" failed to return a correct answer that matches with user's intent question.

Among FREyA's main objectives is to improve understanding of the question's semantic meaning so that FREyA may provide a concise answer to the user's question. For FREyA, in [7] combine the syntactic parsing with ontology reasoning in order to extract the information contained in the user's NL question. Questions such as "Give me the number of rivers in California?" and "count the states which have elevations lower than what Alabama has?" have returned no answer and incorrect answer respectively. Ambiguities of term's semantic meaning have resulted in the incorrect answers to be drawn from the KB. Accordingly, the QA system must be able to correctly interpret all the modifier terms in the user's question so that the returned answer will be successful and accurate. The QA must be able to identify, interpret and quantify the associated modifier term.

Additionally, a different form of question representation may, however, yield a similar answer. For example, the question "What is the capital of Texas?" and "Can you tell me the capital of Texas?"; both forms of questions yield the same answer although they are packed linguistically different. These terms are needed for the treatment by assigning unique evaluation metric for each of the associated modifier terms. In the past, several metrics have been proposed and implemented including UM [17] and relevance feedback (RF) mechanism [15].

User profiling is performed to retain individual user information, experiences, common goals and requirement behaviours. User profiling can characterize the user to a certain style of the question since users mostly have little knowledge of the contents and structures of the knowledge base. Through user profiling, the context of the question may also be emphasized. User profiling can be achieved through UM. In YourQA, it utilizes user modelling technique together with a web search engine to generate answers from a KB [17]. This open-domain interactive QA system provides answers including descriptions and definitions. The user modelling technique basically has the task to filter the

documents in KB and re-rank the answers based on the degree of match with the user profiles, which are based on 3 parameters namely age range, reading level and interests.

Many works also consider RF technique in QA systems, and it is proven that RF can improve the answers ranking process and the performance of QA systems [7], [15], [18]. To improve a QA system, experimented a pseudo relevance feedback for a probabilistic information retrieval system [15]. The model assumed that the document/information which contains the correct class name entities is having more potentiality of relevance than those are not containing them. In other words, although the documents have the correct topic but with no named entity of the expected answers, category will not give a high probability of relevancies. By using named entity parameter, the pseudo relevance feedback can help the users to target the relevant documents at the top rank and eliminate the non-relevant documents more effectively.

Despite their individual strengths and contribution in QA systems mentioned above, a combination of UF and RF has not been manipulated in the area question analysis for QA systems. To fill in this gap, this study presents syntax, and semantic analysis proposes a new formulation strategy for analysing a complex for NL question especially containing a specific term known as “modifier term”. The modifier term is to indicate whether the question needs to execute the comparative, evaluative or other complex question operations. Once the term is identified, the treatment or the strategy is assigned to the associated modifier term. This is as an effort to increase the accuracy of the returned answer to the user NL question by a QA system. Details on the modifier term will be discussed in the next section.

The UM and RF approach acts as a formulation strategy in analysing, interpreting and converting user question into further-processed queries for QA system. Therefore, there are several components that need to be thoroughly understood before a better performance QA system can be formulated. Here, a better performance QA means the system is able to return a correct answer based on user’s intent question. The first essential task is the understanding of the users’ needs and users’ information seeking behaviour. The second essential task is the analysing and processing of the users’ needs expressed in a question (request). Lastly, the third task is providing a strategy for matching of the user’s question to data or information on the document collections or knowledge base [16].

The investigations of this study are embedded in the following research questions:

- How users’ needs and information seeking shall be understood from the user’s NL question?
- How shall users’ needs that are expressed in a question be analysed, interpreted and processed?
- How shall the user’s question be matched for the answers on the knowledge base?

II. MATERIAL AND METHOD

A. The Syntax and Semantic Question Analysis

The study is on the question analysis module which will receive natural language (NL) question that contains modifier terms. The QA system with this semantic question

analysis will return the answer by consulting the gold-standard ontology and knowledge base (KB) taken from Raymond Mooney Dataset [14]. A set of questions used to query the KB is based on Mooney Dataset. The dataset contains 607 annotated user questions with modifier terms from the total of 880 user questions. The Geobase ontology and KB is on US geographical information.

The syntax and semantic question analysis are executed by the Question Analysis Module (QAM) for a prototype of QA system called QAUF (Question Answering system with User Modelling and Relevance Feedback). QAUF involves 2 main modules that are: Question Analysis Module (QAM) and Answer Retrieval Module (ARM). QAM utilizes the combination of user modelling and relevance feedback techniques in analysing and processing the user’s NL question. Whereas, the Answer Retrieval Module (ARM), QAUF adopts the component from an existing ontology-based QA system known as FREyA [7] with an improvement of considering the syntax of the user’s NL question and the semantic descriptions defined by the ontology. The fundamental architecture of the overall QAUF is shown in Fig. 1.

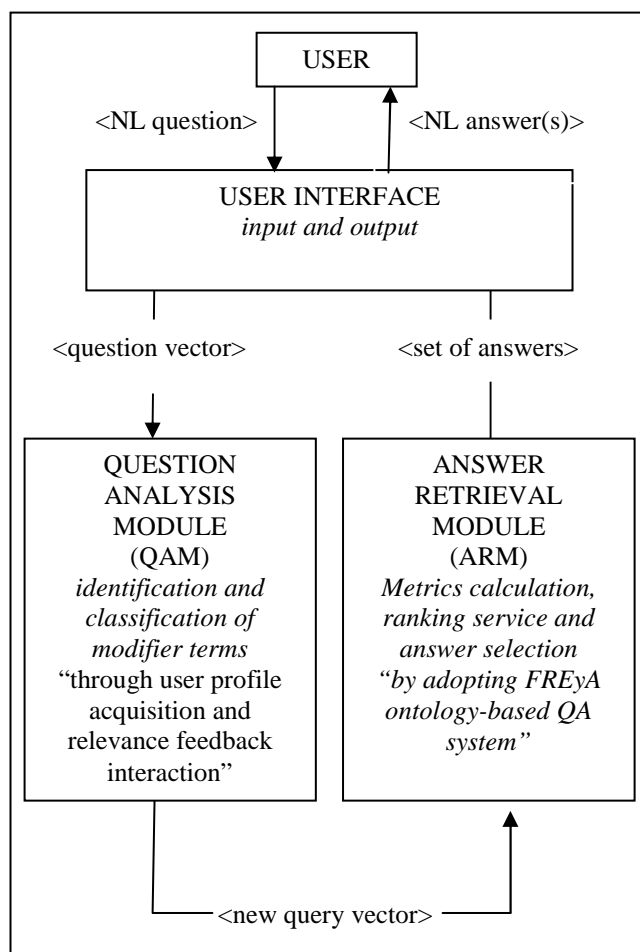


Fig. 1 High level QAUF architecture

In QAUF, user’s NL question is entered into and returned answer to the user through the user interface session. The question containing modifier term is processed in the QAM where identifying, classification and interpretation of

modifier term are performed. The vector space model is used to represent the user's NL question and generalize it into the submitted and modified query. The modified query from QAM will be used in the ARM. Computation and ranking of the candidate answers will be carried out in ARM before the answer can be returned to the user. This paper will only discuss details of QAM in the following sub-section.

B. Question Analysis Module (QAM)

The purpose of Question Analysis Module (QAM) is to analyse and interpret the modifier term in the user's NL question. The output from this module will become the input for the next module, Answer Retrieval Module (ARM) which a correct answer will be returned to the user. The brief steps of QAM processes are as follows:

- Identifying and extracting terms of user question
- Applying user modelling in analysing user question
- Applying relevance feedback in analysing user question

Before details of the QAM is explained and shown, let the general concept of the question analysis be presented. The question analysis concepts are formalized as follows. A user who provides an NL question (Q) and has a specific requirement (U) in retrieving a non-empty set of answer from the knowledge base. A knowledge base that contains a set of information labelled as $I = \{i_1, i_2, \dots, i_n\}$ and a set of $S_i \subseteq I$, which contains information that totally (or partially) relevant to the user's requirement [1-2]. Details on user's requirement will be explained in Equation (2). Thus, to produce S_i , the ontology-based QA system depends on the user requirement and all information in the knowledge base which are represented in the form of function

$$f: (U, I) \rightarrow (S_i, L_i) \quad (1)$$

where L_i is the quality of S_i to the question that can be calculated by using the standard information retrieval measurement.

For example, an NL user's question is "Could you tell me what is the highest point in the state of Oregon?" produces a specific user requirement (U) which intends to retrieve a non-empty set of answer from the knowledge base. In this study, the Raymond Mooney knowledge base is used for question analysis and answer retrieval. Here, in the Raymond Mooney knowledge base contains a set of information labelled as $I = \{texas, oregon, \dots, i_n\}$. A set of $S_i \subseteq I$, is information that totally (or partially) relevant to the user's requirement where a set of S_i considers as potential answers to U. In other words, potential answers to U may consist of S_i, S_j, \dots, S_n . Each of S_i, S_j, \dots, S_n is associated with L_i, L_j, \dots, L_n where L is the quality of S to the question. L can be calculated by using the standard information retrieval measurement, Precision (P).

In this model, U, is a combination of 3 elements such as NL question, user modelling and relevance feedback. This is denoted as follows

$$U = (U_{nl-question}, U_{user-modelling}, U_{relevance-feedback}) \quad (2)$$

Within this context, each element defines as

- $U_{nl-question}$ -is the goal/answer to be retrieved

- $U_{user-modelling}$ -is a set of user interest. It is defined as a set of terms from the user understanding of the domain from knowledge base concept, question context, and language theory. Therefore, this set can be denoted as

$$U_{user-modelling} = \{\text{knowledge base concept, question context, language theory}\}$$

Details and working example of $U_{user-modelling}$ will be presented in the next sub-section.

- $U_{relevance-feedback}$ - is a set of user's feedback given to the system. It is defined as a set of user's domain requirements. In this model, the user's domain requirements are formed from modification of term weight, query expansion, and query simplification. Details and working example of the $U_{relevance-feedback}$ will be presented in the next sub-section.

$$U_{relevance-feedback} = \{\text{modification of term weight, query expansion, query simplification}\}$$

Based on the definition given above, therefore a complete function of the proposed question analysis model is expressed as follows

$$f: (U_{nl-question}, U_{user-modelling}, U_{relevance-feedback}) \rightarrow (S_i, L_i) \quad (3)$$

The user enters the NL question through the QAUF user interface (refer to Fig. 1). User's NL question consists of words which also referred as terms in this study. Terms of user's question are extracted in order to proceed with further processes of finding the correct answer. Terms focus, head focus, modifier term, modified term, focus complement and user profile become input to the QAM [1], [2]. The user's NL question is also extracted its triple which is in the form of <subject, verb, object>. Further details on identifying and extracting these terms of user's question will be described in the following sub-section.

C. Identifying and Extracting Terms of User Question in Question Analysis Module (QAM)

The NL question consists of terms (keywords), and each term is associated with its own weight. The weight of each term is based on the term frequency and inverse term frequency in the knowledge base. This set of terms and weights becomes the question's properties. As in Equation (2), this study uses vector space model to represent the question's properties which consist of a set of <terms, weights>. Question properties must be adequately obtained from user's NL question and knowledge base in order to generate an executable query for retrieving the relevant answer [1], [2]. However, user's NL question becomes ambiguous when it contains modifier term. This is because question with modifier terms requires QA systems to comprehend and evaluate based on its syntax and semantic dimensions before the relevant answers can be yielded. Thus, additional information is demanded in disambiguation which can be achieved by enriching the question's properties. To enrich the question properties, an additional set of new terms must be inserted to the question. In this study, user modelling and relevance feedback are used to generate new

terms. This approach is known as a modification of query which resulting in a new query vector.

As mentioned previously, to yield the relevant answers modifier term and other appropriate terms need to be extracted during question analysis. These terms are referred as lexical terms which comprise focus term, head focus term, focus-complement term, modifier term and modified term. These lexical terms are extracted according to the heuristics-based, which depends essentially on the syntactic form of the question. The lexical terms are extracted and syntactically analysed using the Natural language processing (NLP) tools specifically on the text processing tools. These are the Stanford Tokenization, Stanford Part-of-Speech (POS) tagger and Stanford Syntactic Parser. The five lexical elements considered in this study are explained in [1], [2]. Heuristics used in determining the lexical terms will be explained in the next following section.

As for an example, the user's NL question of "What are the cities of the state with the highest point?" is tokenized, assigned its part-of speech and parsed as

User's NL question:

"What are the cities of the state with the highest point?"

Tokens: 12

(what-1)
 (are-2)
 (the-3)
 (cities-4)
 (of-5)
 (the-6)
 (state-7)
 (with-8)
 (the-9)
 (highest-10)
 (point-11)

POS Tagging:

what/WP
 are/VBP
 the/DT
 cities/NNS
 of/IN
 the/DT
 state/NN
 with/IN
 the/DT
 highest/JJS
 point/NN
 ?/.

Parse:

(ROOT
 (SBARQ
 (WHNP (WP what))
 (SQ (VBP are)
 (NP
 (NP (DT the) (NNS cities))
 (PP (IN of)
 (NP
 (NP (DT the) (NN state))

(PP (IN with)
 (NP (DT the) (JJS highest) (NN point))))))
 (. ?))

Typed Dependencies:

root(ROOT-0, What-1)
 cop(What-1, are-2)
 det(cities-4, the-3)
 nsubj(What-1, cities-4)
 prep(cities-4, of-5)
 det(states-7, the-6)
 pobj(of-5, states-7)
 prep(states-7, with-8)
 det(point-11, the-9)
 amod(point-11, highest-10)
 pobj(with-8, point-11)

The part-of-speech name abbreviations are based on the Penn Treebank tag set for English taggers. The parsed result of the user's NL question also includes the Stanford typed dependencies. The Stanford Typed Dependencies give a representation of grammatical relations between terms in user's NL question. This typed dependency is formed in triplet forms. Terms are parsed and based on their grammatical structures and relations of the given user's NL question. Then, these terms are categorized and labelled for their appropriate lexical elements which have defined in [1], [2]. Terms categorization is done based on the English syntactic rules. Rules depend on the grammar equations and typed dependencies of the parsed result. In determining the lexical terms, syntactic heuristics are used in this study. Fig. 2 and Fig. 3 present the syntactic heuristics used to identify and categorize the modifier and modified terms respectively.

```
//To find modifier term
1. Start
2. Repeat until end of typed-dependencies-list
   i. If (amod is found) || (advmod is found) ||
      (predet is found) then
      a. Get token for terms of amod ||
         advmod || predet
      b. Get POS for terms of amod || advmod
         || predet
      c. modifier term ← term of ((JJ || JJR ||
         JJS || CD) of amod) || ((RB || RBR ||
         RBS of advmod) || (PDT of predet)
3. End
```

Fig. 2 The syntactic heuristics for modifier terms

```
//To find modified term
1. Start
2. Repeat until end of typed-dependencies-list
   i. If (amod is found) || (advmod is found) ||
      (predet is found) then
      a. Get token for terms of (amod) ||
         (advmod) || (predet)
      b. Get POS for terms of (amod) ||
         (advmod) || (predet)
      c. Modified term ← term of ((NN ||
         NNS)of amod) || (( VBN of
         advmod)) || ((NN || NNS) of predet)
3. End
```

Fig. 3 The syntactic heuristics for modified terms

As QAUF receives user's NL question, the basic operation of QAM is to extract the lexical terms. Then, the user's NL question is also extracted its triple which is in the form of <subject, verb, object>. Based on the parsed result and extracted lexical terms, the triple extracted for the user's NL question in the example above is <city, are, state>. However, in this study, the user's question contains modifier and modified terms, and these terms have the dependency on the object of the triple i.e.: state - highest - point. This requires additional information on the semantic description of the terms as well as on the appropriate operation execution for answer finding. Therefore, user modelling is applied.

D. Applying User Modelling (UM) in User's Question

The aim of using User Modelling (UM) is to increase the accuracy by providing additional information in order to filter the potential relevance answers from the knowledge base. In the stage of UM, the model attempts to capture user's interest based on 3 aspects that can be labelled as

$$U_{\text{user-modelling}} = \{\text{knowledge base concept, question context, language theory}\}$$

These aspects shall be a set of class concept and properties of the knowledge base and the question context using language model. In capturing the user's interest, the question analysis module (QAM) utilizes two from five methods reported by [19]. They are default assumptions and assumption contribution from the system. This is to minimize the user involvement during this phase. Based on these two selected methods, the model attempts to capture user's interest on 3 aspects such as knowledge base concept (default assumption method), question context (default assumption method) and language theory (assumption method). Therefore, the profile of user's interest can be denoted as

$$U_{\text{user-modelling}} = (U_{\text{knowledge-base concept}}, U_{\text{question context}}, U_{\text{language theory}}) \quad (4)$$

For the knowledge base concept, question context and language theory aspects, QAM consider instances to the class concept, concepts or properties of the knowledge base or any term from the user's question structure [20]. For

example, name of city, name of river and name of mountain. The initial user profile is auto-generated based on these aspects captured. The auto-generated user profile is used to generate a new query for retrieving the relevant answer. The initial auto-generated user profile will be saved as a user profile, which can be obtained when the user decides to post another question to the QA system.

Based on the example used previously, the user's NL question: "What are the cities of the state with the highest point?" gives $U_{\text{nl-question}} = \{\text{city, state, highest, point}\}$. The items of $U_{\text{nl-question}}$ are terms of the lexical elements. User question also has the triple format as <city, are, state>. This triple is mapped to the concepts and relationships from the knowledge base. As for the subject, the city is mapped to any corresponding class. For the verb, are is mapped to any corresponding property in the knowledge base which has a relationship with the class city. The value of the object in the triple determines the value of subject in the triple which becomes the answer for user's question. Since the user's question (refer to $U_{\text{nl-question}} = \{\text{city, state, highest, point}\}$) contains modifier and modified terms and these terms have the dependency on the object of the triple. This demands additional information to semantically interpret the modifier and modified terms in order for QAUF to return the answer to user's intent question.

Based on $U_{\text{nl-question}} = \{\text{city, state, highest, point}\}$, QAM in QAUF performs the initial user profile suggestion which generates the items of $U_{\text{user-modelling}} = \{\text{isCityOf, State, isHighestPointOf, hasHighestPoint, HiPoint, city, state, highest, point}\}$. These are the additional information require to further process the user's question. To generate the user profile, a string similarity and a semantic similarity technique are used. First, QAM consults the ontology and knowledge base to map items of $U_{\text{nl-question}}$. Items of $U_{\text{nl-question}}$ require being referred to WordNet for term's semantic information if they fail to be mapped with the ontology and knowledge base. Also, the Modifier Lookup is referred for an interpretation which reflects an appropriate operation execution of the modifier term.

Referring to the user's question in the example above, items of $U_{\text{nl-question}} = \{\text{city, state, highest, point}\}$ manage to be mapped with the ontology, knowledge base and Modifier Lookup that generates $U_{\text{user-modelling}} = \{\text{isCityOf, State, isHighestPointOf, hasHighestPoint, HiPoint, city, state, highest, point}\}$. $U_{\text{user-modelling}}$ is the user profile. Based on this user profile, a new query (Q') is constructed for user's NL question (Q). The defined concepts or terms of the question in $U_{\text{user-modelling}}$ are represented in a vector space model as follows

$$Q' = \{(c_1, w_1), (c_2, w_2), \dots, (c_n, w_n)\} \quad (5)$$

where c represents the classes, properties or instances of the knowledge base which is based on term of user's question. c may also represent question terms. w is the weight of c. In the next sub-section, the phase 2 of QAM which to applying relevance feedback in the user's query (Q') is presented.

E. Applying Relevance Feedback (RF) in User's Query

In phase 2, relevance feedback is applied to the user's query (Q') that corresponds to the user's question. If

necessary, QAM in QAUF allows the user to modify the weight associated with the question term. This will contribute to the formation of the new query vector (Q''). As discussed earlier, to avoid ambiguities in user's question terms, QAM acquires additional information through inserting new terms in regards to the question entered. Similar as in Phase 1 (Applying UM), new terms insertion is accomplished by consulting WordNet, Knowledge Base, Ontology, and Modifier Lookup. As a result of applying RF, another new query vector (Q'') is produced. In the next paragraph, the concept of applying relevance feedback in QAM is explained in details.

In applying relevance feedback, the question analysis model considers 3 aspects of modifying the query representation and can be defined as $U_{\text{relevance-feedback}} = \{\text{modification of term weight, question expansion, question simplifying}\}$. QAM provides modification or reformulation of the weight of each term that exists in the query if the user finds it the best for QAUF to return an answer to user's intent question. The modification is done based on the user requirement (U). For example, in the user profile, the term `hasHighPoint` has 0.0004 term weight. However, the user may modify the term weight of `hasHighPoint` into the possible value ranges 0.0001 till 0.9999, depending on the user requirement.

Also, the user can expand the question by inserting new terms such as adding synonyms of head focus, modifier term, modified term, focus complement and category of expecting answers. The question expansion is constructed by exploiting the knowledge base, WordNet, Modifier Lookup, ontology and user requirement specification or domain. Thus, for the user's NL question of "What are the cities of the state with the highest point?", new terms are added as the semantic interpretation and relation of modifier and modified terms. By consulting the WordNet and Modifier Lookup, the term `high` and `height` are added since they carry possible semantic meanings to the modifier term `highest`. The modified term, the `point` is already mapped to the ontology and knowledge base in the phase 1 which resulted in term `isHighestPoint` and `HiPoint` to be included in the user profile. When ontology and knowledge base are referred to the semantic relations, terms `Mountain` and `hiElevation` are inserted as new terms. The user may choose these terms (i.e.: `Mountain`, `hiElevation`, `height`) that best fit with the user's intent question.

In phase 2 of RF, another new query vector (Q'') is produced. This new query vector (Q'') can be generalized as a boolean combination that contains the original $t_1, t_2, \dots, t_n \in T$ from user's question (Q) plus additional semantics terms correspond to the knowledge base, WordNet, Modifier Lookup, ontology and user requirement domain. When RF is implemented to the Q by modifying the question (modification of weight, adding new modifiers, simplifying the question), this produces Q'' which denotes as:

$$Q'' = \{q_1'', q_2'', q_3'', \dots, q_n''\} \quad (6)$$

Thus, the set of solutions can be represented as follows

$$q_i'' = \{(T_{1,i}, \Delta W_{1,i}), (T_{2,i}, \Delta W_{2,i}), \dots, (T_{n,i}, \Delta W_{n,i})\}, \{(T_{n+1,i}, W_{n+1,i}), (T_{n+2,i}, W_{n+2,i}), \dots, (T_{k,i}, W_{k,i})\} \quad (7)$$

where, ΔW represents the change of weight value, and $T_{n+1,i}, T_{n+2,i}, T_{k,i}$ are the new inserted terms that associated with their weight $W_{n+1,i}, W_{n+2,i}, W_{k,i}$ respectively and $i = 1, 2, 3, \dots, n$ and $k > n$.

Based on the example of the user's question given above

User's NL question:

"What are the cities of the state with the highest point?"

Triple format: <city, are, state>

modifier term: highest; modified term: point;

where the term `state` has direct dependency with both modifier and modified terms (`state - highest - point`). The user may choose new terms (i.e.: `Mountain`, `hiElevation`, `height`) that best fit with the user's intent question.

In RF phase, the user may select terms that are best fit to the user's intent question. Therefore, in this example, the user has chosen the new term `mountain` and `height` that best describe the modifier and modified terms; `highest` and `point`. Apart from that, the user is also allowed to input any new term(s)/word(s) that may contribute additional information to any possible new q_i'' . Any possible new q_i'' containing term `hiElevation` will be disregarded.

To determine the best solution to be used, the initial NL question Q is compared with all solutions Q'' and the similarity scores are computed. The similarity score formula used is as follows

$$\text{Similarity}(Q, q_i'') = \sum_{m=1}^k (W_m \times W_{im}) \quad (8)$$

Based on Equation (6), the best possible of q_i'' (that is the nearest similar to Q) is selected from all possible Q''. Here, q_i'' that has the highest similarity score is recognized as nearest similar to Q which is 0.0129 (q_6''). If Q_{new} is the new query vector that is submitted to the ARM, thus Q_{new} can be denoted as follows

$$Q_{\text{new}} = (\text{rule}, \text{highest}(q_i'')) \quad (9)$$

where `rule` is the rule obtained from the Modifier Lookup during the stage of applying UM technique and will be applied to the query containing the modifier and modified terms and `highest(q_i'')` is the nearest similar query vector that generated in the stage of applying RF technique.

III. RESULTS AND DISCUSSION

In this section, experiments are done to quantify the returned answers of user's NL questions (Q) which contain modifier terms. The question analysis model implemented in QAUF shall return the correct answers for the user's NL question. The user's question with the modifier term is mapped with the Modifier Lookup to get the appropriate rules for the query execution operation. Additional information is provided through user model (i.e.: based on user profile) and relevance feedback. Experiments which combine the auto-generated user profile and relevance feedback are performed.

Table 1 consists the 10 questions from Raymond Mooney dataset is used to depict the similarity scores acquired in this study. Table 2 shows the similarity score for these 10 sample users' questions. Table 3 shows the performance of a question analysis model in processing the 10 sample users' questions. The results are evaluated in terms of F-Measure and average precision (AP) of the returned correct answers

TABLE I
10 SAMPLES NL QUESTIONS

Num	Question
11	'Could you tell me what is the highest point in the state of Oregon?'
22	'How big is Alaska?'
33	'How tall is the highest point in Montana?'
44	'How large is the largest city in Alaska?'
55	'How many major cities are in Arizona?'
66	'Name all the rivers in Colorado?'
77	'What are the cities of the state with the highest point?'
88	'What are the highest points of all the states?'
99	'What city has the most people?'
110	'What is the biggest city in Arizona?'

TABLE II
SIMILARITY SCORES OF Q AND Q''

Num	$Sim(Q, Q'')$	Highest $Sim(Q, Q'')$
1	0.0317	0.0317
	0.0023	
2	0.0028	0.0028
	0.0027	
3	0.0396	0.0396
	0.0002	
	0.0205	
	0.0206	
	0.0002	
4	0.0043	0.0043
5	0.0042	0.0042
6	0.0012	0.0012
7	0.0068	0.0129
	0.0033	
	0.0035	
	0.0039	
	0.0105	
	0.0129	
8	0.0033	0.0035
	0.0035	
9	0.0046	0.0089
	0.0089	
10	0.1331	0.1331

As shown in Table 1, the users' NL questions (Q) that are selected from Raymond Mooney Geoquery dataset contain modifier terms. These modifier terms are referred to the Modifier Lookup to determine the operation of query execution. Applying UM and RF to Q produces Q' and Q'' respectively. Table 2 shows the similarity scores of user's NL questions, Q and newly produced queries, Q'', after applying UM and RF. Some of the user's NL questions obtained several similarity scores when user's question terms are compared with the underlying references (i.e.: WordNet,

Modifier Lookup, Ontology and Knowledge Base). The highest similarity score is recognized as the nearest similar to the original user's NL question, and it will be submitted to the Answer Retrieval Module for the answer. The new query Q'' relaxes the ambiguity of modifier terms exist in the user's question by providing additional information. The highest similarity score of the new query Q'' will be chosen as the most similar query with the user's NL question, Q. This new query will be submitted to the Answer retrieval Module. In this experiment, the accuracy of the returned answer is evaluated based on the correct answer yielded by QAUF in answering the user's intent question. Table 2 lists similarity scores for 10 samples Q.

Table 3 shows the results on P, F-measure and AP for 10 sample questions shown in Table 1.

TABLE III
PERFORMANCE OF APPLYING UM AND RF ON 10 USERS' QUESTIONS

Num	R	Returned Accurate Answer (P)	F-Measure	P	AP
1	1.0	1.0	1.0	1.0	0.1
2	1.0	1.0	1.0	1.0	0.1
3	1.0	1.0	1.0	1.0	0.1
4	1.0	1.0	1.0	1.0	0.1
5	1.0	1.0	1.0	1.0	0.1
6	1.0	1.0	1.0	1.0	0.1
7	1.0	1.0	1.0	1.0	0.1
8	1.0	1.0	1.0	1.0	0.1
9	1.0	1.0	1.0	1.0	0.1
10	1.0	1.0	1.0	1.0	0.1

As can be seen from Table 3, UM and RF implemented in question analysis module for QAUF shows 100% good indications in P, F-measure and AP of 10%. These results are obtained for 10 users' questions as shown in Table 1.

IV. CONCLUSION

In this paper, the syntax and semantic analysis for user's NL question have combined the UM and RF techniques in QAM of QAUF. The result of combining these two techniques manages to solve issues that were discussed. This approach becomes a new strategy to semantically interpret the given user's NL question. A series of experiments were done to evaluate the effectiveness of syntax and semantic question analysis in terms of F-measure and AP. User's NL questions are from Raymond Mooney Geography dataset which questions containing modifier terms are selected for the experimentations. User's NL question is analysed, interpreted and transformed into a new query. QAUF demonstrates a good precision percentage in returning relevant answers for each query with 94.7% F-measure and AP. The finding of this study demonstrates that the syntax and semantic analysis [21] which combines UM and RF QAUF has a good AP percentage in returning relevant answers for each NL question.

In the near future, this study aims to categorize the modifier Lookup by considering the knowledge base context and the pragmatic of the user's NL question.

ACKNOWLEDGMENT

Special thanks to Universiti Sultan Zainal Abidin (UniSZA), Universiti Putra Malaysia (UPM) and Management of Faculty of Informatics and Computing, UniSZA.

REFERENCES

- [1] S. I. A. Saany, A. Mamat, A. Mustapha, and L. S. Affendey, "A strategy for question interpretation in question answering system," *International Journal of Computer Science and Telecommunications*, vol. 4, pp. 38-43, 2012.
- [2] S. I. A. Saany, A. Mamat, A. Mustapha, and L. S. Affendey, "Incorporating user modeling and relevance feedback in question analysis model," in *Proc. IEEE CCSII'13*, 2013, p. 113.
- [3] P. Atzeni, R. Basili, D. H. Hansen, P. Missier, P. Paggio, M. T. Pazienza, and F. M. Zanzotto, "Ontology-based question answering in a federation of university sites: The MOSES case study," in *Proc. ICANLIS'04*, 2004, p. 413.
- [4] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Y. Lin, S. Maiorano, G. Miller, and D. Moldovan, "Issues, tasks and program structures to roadmap research in question & answering (Q&A)," in *Proc. DUCRD'01*, 2001, p. 1.
- [5] B. C. Pal, P. Bhaskar, and S. Bandyopadhyay, "A rule based approach for analysis of comparative or evaluative questions in tourism domain," in *Proc. KRAQW'11*, 2011, p. 29.
- [6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A framework and graphical development environment for robust NLP tools and applications," in *Proc. AMACL'02*, 2002.
- [7] J. J. González, R. Florencia, H. J. Fraire, R. A. Pazos, L. C. Reyes, and C. Gómez, "semantic representations for knowledge modeling of a natural language interface to databases using ontologies," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 6, pp. 28-42, May 2015.
- [8] A. B. Abacha and P. Zweigenbaum, "MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies," *Information Processing and Management*, vol. 51, pp. 570-594, Sep. 2015.
- [9] J. Fink and A. Kobsa, "User modeling for personalized city tours," *Artificial Intelligence Review*, vol. 18, pp. 33-74, Sep. 2002.
- [10] A. Sayed and A. Al-Muqrishi, "CASONTO: An efficient and scalable arabic semantic search engine based on a domain specific ontology and question answering," *International Journal of Web Information Systems*, vol. 12, pp. 242-262, Jun. 2016.
- [11] R. Iqbal, M. A. A. Murad, M. H. Selamat, and A. Azman, "Negation query handling engine for natural language interfaces to ontologies," in *Proc. ICIRKM'12*, 2012, p. 294.
- [12] N. R. Lim, P. S. Dizier, and R. Roxas, "Some challenges in the design of comparative and evaluative question answering systems," in *Proc. WKRAQ'09*, 2009, p. 15.
- [13] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE ICCV'15*, 2015, p. 2425.
- [14] MooneyData. (1994) Geoquery data. [Online]. Available: <http://www.cs.utexas.edu/~ml/nldata/geoquery.html>
- [15] L. A. Pizzato, D. Molla, and C. Paris, "Pseudo relevance feedback using named entities for question answering," in *Proc. ALTW'06*, 2006, p. 83.
- [16] S. Quarteroni, "Personalized question answering," *TAL*, vol. 51, pp. 97-123, 2010.
- [17] S. Quarteroni and S. Manandhar, "Designing an interactive open-domain question answering system," *Journal Language Engineering*, vol. 15, pp. 73-95, 2009.
- [18] V. Lopez, V. Uren, E. Motta, and M. Pasin, "AquaLog: An ontology-driven question answering system for organizational semantic intranets," *Journal of Web Semantics*, vol. 5, pp. 72-105, Jun. 2007.
- [19] W. Wahlster and A. Kobsa, *User Models in Dialog Systems*, ser. User Models in Dialog Systems. Berlin, Germany: Springer, 1989.
- [20] I. M. Yassin, A. Zabidi, M. S. A. M. Ali, N. M. Tahir, H. A. Hassan, H. Z. Abidin, and Z. I. Rizman, "Binary particle swarm optimization structure selection of nonlinear autoregressive moving average with exogenous inputs (NARMAX) model of a flexible robot arm," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, pp. 630-637, Oct. 2016.
- [21] M. N. M. Nor, R. Jailani, N. M. Tahir, I. M. Yassin, Z. I. Rizman, and R. Hidayat, "EMG signals analysis of BF and RF muscles in autism spectrum disorder (ASD) during walking," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, pp. 793-798, Oct. 2016.