Journal of Theoretical and Applied Information Technology
30th June 2017. Vol.95. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTOMATED SEMANTIC QUERY FORMULATION USING MACHINE LEARNING APPROACH

[1]RABIAH A.KADIR, [2]ALIYU RUFAI YAURI
[1] Institute of Visual Informatics, Universiti Kebangsaan Malaysia
[2]Department of Computer Science, Kebbi State University of Science and Technology
E-mail: [1]rabiahivi@ukm.edu.my, [2]rufaialeey@yahoo.com

ABSTRACT

Search engines such Yahoo and Google among others has played significant role Web data access. However these search engines has limitations. These search engines are based on a keyword search which lacks semantics in the retrieval process. To cope with the Limitations of current search engines, Semantic Web was introduced. Semantic Web enables retrieval of data on the Web semantically. In semantic Web, data is standardised in a format that enables retrieval of such data semantically. But Semantic Web also has challenges where retrieval requires complex structured query such as SPARQL which is not simple are using Google like natural language query. This paper presents an approach of automatic semantic query formulation that enables retrieval of semantically structured data using natural language. The proposed approach is based on using machine learning and the result has shown improvement of 17.4% compared to existing approach in FREyA in terms of effectiveness formulated natural language queries to structured query.

Keyword: *semantic Web, Machine learning, ontology, Quran.*

1. INTRODUCTION

Although current search engines usually return varied information related to search queries, most of these returned results are irrelevant. In most cases users need to navigate through several pages before finding relevant information. Sometimes a particular query may not be answered by a single document, instead several documents needs to be navigated before finding relevant information [1]. These challenges are because the current Search Engines are base on traditional keyword search. The search engine doesn't understand the meaning of user query. For example, a query "what is the cost of a Jaguar car", current search engine such as Google will return information about jaguar cars, jaguar as animals, or as products, among other things are returned without considering the true meaning of the query, which is *Jaguar car*.

To overcome the shortcomings of the traditional keyword based search systems, the concept of the Semantic Web was introduced by the W3C consortium. Semantic Web, in other words a web of linked data, is an extension of the current version of the Web whereby information is given a well-defined meaning to enable human and computers to easily work together. Semantic Web models the meaning of information on the web, as well as applications and services, so as to discover, annotate, process and publish data that is encoded in them[2]. Semantic Web enables facilitation of semantic searches on data, where computers understand user's query intention and retrieve corresponding results based on matching concepts rather than keywords[3]. In Semantic Web, data is represented into Resource Description Framework (RDF) format. RDF is a W3C recommended language for representing data on the s web. RDF uses ontology to transform data into graphical triple form representation. (Subject, Predicate, Object)

In simple terms ontology are objects that exist in a particular domain and the relationships between the objects. In spite of the success of Semantic Web in the retrieval process, the challenge remains querying data represented in RDF structure require structured complex query such as SPARQL which requires user to be familiar with how to construct the query. Recent studies show that users prefer using natural language to structured query language [4]. Users prefer natural language query to structured query because natural language query systems hide the complexity of the structured query. Although quite a number of works has been reported previously in that regard, due to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the complex nature of natural language, the area remains subject to research. Most of the current works are semi-automatic approach, where semantic query formulation is base human and Computer interaction.

The approach in this paper improved the semantic query formulation process from semi-automatic to automatic. Our approach uses machine learning to automatically formulate user's natural language query to structured triple representation. The semantically formulated triple is then used to generate a SPARQL query which is matched against the knowledge base to retrieve relevant results semantically.

The proposed approach has few features that differentiate it from the existing approaches. Firstly, the system allowed users to query using either single or multiple sentence query where most current systems are based on single sentence query. Secondly, is based on automatic semantic formulation of natural language query without human intervention. Thirdly, the system propose hybrid-automated disambiguation process that enables automatic disambiguation of ambiguous query words using WordNet lexical dictionary and ontology equivalent assertion of for words that are lacking in WordNet. Fourthly, the proposed system, provide suggestion in case the system fails to automatically formulate natural language query to structured triple representation with or without the presence of ontology concepts in the natural language query.

The rest of this paper if organized as follows. Section 2 provides related work. Section 3 contained an over view of the proposed system. Section 4 presents an exclusive evaluation and analyses of the proposed system. And section 5 presents collusion and future direction of the research.

2. RELATED WORK

Semantic Web was designed as a new generation of the current web, where data is given clear meaning to enable computers to easily work together [5]. The main building block of the semantic web is ontology, which transforms web content into a machine-readable format that can be manipulated [6]. Ontology, in other words Web Ontology Language (OWL), is commonly defined as formal and explicit specifications of shared conceptualization. Formal signifies ontology as a machine-readable format. Whereas, the concepts or entities used are explicitly described, shared, and displayed, ontology is concept that captures knowledge in a widely acceptable standard, and its conceptualization reflects ontology as a notion that identifies entities in the real world [7]. Ontology is modelled into machine-understandable format, known as RDF, for computer applications to use when making inferences [8]. RDF stands for resource description framework which was mainly a practical rule language for computers to understand, manipulate and share data [9].

Semantic search enables computers to think, reason, manipulate RDF data and provide humans with the information they need in the way that they need it [10] using structured query. A structured query involves the use of formal structured rules to generate a query in order to use it for a retrieval process (Tannier, Girardot, Mathieu, & Saint-étienne, 2002).Figure 1 shows an example of SPARQL query language.

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
        SELECT ?x ?name
WHERE { ?x foaf:name ?name }
```

Figure 1: Example of structured language (SPARQL)

Figure 1 is a SPARQL query which natural language means "select all names that are in friend of friend data (foaf). SPARQL has complex syntax which requires prior knowledge how query should be constructed. In order to enables the user of preferred natural language to query structured data, the concept of semantic query formulation was introduced. The goal of semantic query formulation is to assist users by formulating their natural language query into a formal structured.

In recent years, several approaches on semantic query formulation researches have been presented. The approaches categorized into manual, semi-automatic and automatic semantic query formulation approach.

Manual semantic query formulation is mostly template based where user is required to semantically formulate structured query manually. In this approach, user is required to be familiar with any of the structured query syntax of the structured query such as SPARQL syntax or have knowledge of how the RDF data is represented in the knowledge base in order to retrieve data from the knowledge base. Ontology editors such as Protégée and some query editors like Virtuoso SPARQL, Flint SPARQL Editor, and Drupal SPARQL Query Builder among others are systems that enable users

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to manually formulate a formal query language and retrieve knowledge from the knowledgebase. Another query editor that allows users to manually construct formal queries was presented in the work of [12]. Manual semantic query formulation approach is complex and time consuming because user needs to learn the complex syntax of the structured query language.

Semi-automatic approaches were presented to enhance the capabilities of the manual based approach. In semi automatic approach, user and machine are involved in the semantic query formulation process. Semi-automatic systems could be template-based, browse like systems or combination of both template and browse like system. In Template base, user is presented with template where he perform preliminary filling of form such as in [13][5][14][15][16] [17]. These template base query formulation approach presents to user with predefined query templates from where they choose to semantically formulate the structured query.

Search and browse base systems improve the templates based approach such as works in TAP [18] and [19]. The main concept of TAP is to enable users to either use the browsing capability provided by the system or search for the information they need. The search mechanism accepts user input in textual form and returns all resources whose title properties contain the text.

Although the above-mentioned approaches offer browsing mechanisms for knowledge in the knowledgebase, determining the right concept for the systems using the posed search query is not straightforward [5]. There ambiguity problem as users may be misled by the system to assume their intended query does not exist in the system, when in fact a different vocabulary is used by the system, such as synonyms in the underlying knowledgebase.

CINDI proposed a form-base query formulation approach where a user poses their query through filling in a form presented by the system and solve ambiguity problem [20]. The system incorporated lexical dictionary WordNet to create a list of hyponyms and synonyms for each relationship and attribute name. ORAKEL is another form based system that semantically formulates a user's natural language query into a structured query with ambiguity solving provision [21].

[22] Present a work that transforms natural language queries into structured queries based on the user interaction approach. The system offers a mechanism that allows users to semantically query the knowledgebase using natural language.

FFQI presented a query formulation approach for retrieving structured data from database [23]. The system is designed to accept natural language queries based on a semantic graph model. A user is presented with an interface that enables them to make some selections that the system uses in formulating a query by using probabilistic popularity measures. In this system, the disambiguation of the user query is done based on ranking technique. The semantic graph is a model for a relational database that is comprised of nodes as relationships, and links are represented as the joins between nodes. When users input a natural language query, the popularity of the nodes and their link is used for the formulation and ranking of the query.

Although the semi-structured semantic query formulation system reduces complexity of manual approach, users need to interact with the system before they can retrieve from the knowledgebase. This is still hectic and time consuming with subject to human error. Today users a better user friendly system such as a Google-like search mechanism where they can easily make a query and receive an answer without participating in the retrieval process. Therefore researchers intensify effort for automatic approach where the system does the semantic query formulation automatically.

Various researchers have proposed automatic semantic query formulation approaches. Although these proposed automatic approaches has shown significant improvement over the semi-automatic query formulation approach, most of these systems are still not fully automated. They mostly still involved users for the semantic query formulation process. Manually or semi-automatic semantic query formulation is tedious and time consuming compared to automatic approach.

AquaLog [24] presented an automatic approach. The system is a portable Natural language interface that enables users to query the knowledgebase using natural language. The query is then formulated into a structured query and matched against the knowledgebase for retrieval. NLP-Reduce is another approach based on automatic semantic query formulation that transforms a user's natural language query into a structured query [25]. The core part of the system is the query generator which is accountable for creating SPARQL queries given the words and the lexicon extracted from the knowledgebase, where users are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

able to enter keywords or full sentences for querying the knowledgebase**.**

QuestIO formulates user's natural language query to a structured query SeRQL automatically [26] . However, the system is based on small fragment queries, which are able to work with ill-formed or incorrect sentences.

PowerAqua [27] supports the transformation of a user's natural language query into structured form automatically. Power Aqua has the advantage of being domain independent, where user queries don't have to target specific domains. User queries can be formulated to retrieve information from semantically structured data on the web. However one of the drawbacks of Power Aquais that the system is limited to single sentence queries.

AutoSPARQL is a query formulation interface that formulates user queries to SPARQL query language [28]. The system is based on a supervised machine learning approach that learns about a user's intended query based on user interaction. The system is also based on small fragment queries and therefore cannot cope with multiple sentence queries. Another drawback of this system is that when the system is unable to find information relevant to the user query it simply does nothing, which may require users to start again from the beginning.

DENNA is another semantic query formulation system mainly designed to transform natural language queries into structured queries [29]. The system has limitations in that when vocabularies which differ with that in the knowledge base are used in queries, the system will not be able to identify them. No provision is made to attempt to search for synonyms in case a user uses different vocabulary from that which exists in the knowledgebase.

The MyAutoSPARQL system [30] is another work that attempt to automatically formulate a user's natural language to a structured query based on the technique of rewriting a NL query.

The above automatic query formulation system has some limitations. These systems simply fail when it is not able to answer user queries. This will require user to start from the beginning if they intend to continue with the search process which may end up frustrating users since they may need to re-write their query several times before they obtain the desired result. Therefore several researches was presented to cope with this issues

Work in [31] is a natural language interface for querying ontologies where the system attempts to automatically semantically formulate natural language queries into structured queries. The system provides the user with a clarification dialogue in case the system fails to answer the query. In FREyA, if the system is unable to successfully generate a triple, a clarification dialogue appears for a user to disambiguate concepts and manually map the concepts with relationships through system-based suggestions provided by the system.

However, the manual, semi-automatic and automatic query formulation systems we have mentioned so far are designed based on small fragment queries, and in some cases, such as an Islamic domain where users ask queries in multiple sentences; these systems may not be able to answer such queries.

SWSNL is a work in [32] was proposed to go beyond phrase or single sentence query. SWSNL works on semantic query formulation that enables users to query the knowledgebase using natural language in a phrase, single sentence or multiple sentences. But there is also no clear provision for disambiguation in there system, such as using a WordNet lexical dictionary to check synonyms of words in case different words were used from those that exist in the knowledgebase, or asking users to make clarifications in case the system is unable to answer their queries. Additionally, although the system goes beyond small fragment queries by providing flexibility in querying using keywords, phrases, single sentences or multiple sentences, it does not have much functionality, like FREyA where the system doesn't just fail when it fails to answer the user. SWSNL does nothing when the system fails to answer the user, which may require users to continue re-writing their queries, which may waste their time, and in the end frustrate the user.

Based on the reported researches presented in this review section, we can say that research should be intensified, towards finding an effective automatic semantic query formulation such as developing a system that will accept paragraph length query as against phrase and single sentence query in the current approaches. Our approach in this paper solves this problem by proposing paragraph length query automatic semantic query formulation approach. Secondly, automatic semantic query formulation systems should not just do nothing when the system fails to answer user queries as in SWSNL. Our automatic suggestion

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

approach allow user either reformulate his query to get automatic suggestion approach.

Thirdly, in terms of disambiguation of ambiguity on user's query, our approach is able to deal with the limitations of WordNet by combining WorldNet and Ontology equivalent assertion to solve ambiguity.

3. AUTOMATED SEMANTIC QUERY FORMULATION USING MACHINE LEARNING APPROACH

The research adapted Statistical Machine Learning Technique for semantic query formulation complex natural language query. The main objective of the research is to semantically formulate natural language to structured triple representation (Subject, Predicate, Object).

The proposed approach accepts simple or complex sentence natural language queries. In this approach the queries are parsed through several linguistic processing, supervised statistical learning technique using N-gram maximum likelihood estimation to semantically transform natural language query to triple representation. In our approach, when system fails to semantically formulates natural language to triple representation, instead of just failing as in previous systems, the system provides options for the user to either reformulate his query or get some suggestion from the system. The suggestion provided by the system gives some possible triple representations of the query. It is important to note that existing system provide suggestion to the user such as in FREyA, but these suggestions are based on the presence of ontology concepts in the query. Therefore when there is no ontology concept in the query, the system will not be able to provide any suggestion to the user. In this research, the system based suggestion is proposed that is able to provide suggestion the user with or without ontology concept in the query. In terms of ambiguity which is almost inevitable when dealing with natural language, the proposed approach, solve ambiguity using WordNet lexical dictionary and ontology equivalent assertion.

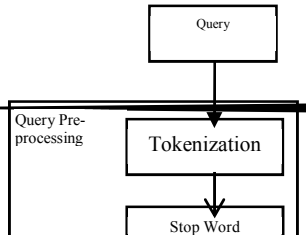

Figure 2: Framework of Semantic Query Formulation

Figure 2 shows the framework of the proposed automated semantic query formulation based on statistical machine learning technique. It shows the modules involved and the process of automated semantic query formulation using a Statistical Machine Learning Technique.

The framework for the proposed automated semantic query formulation based on statistic machine learning technique is presented in figure 2 which comprises of two modules:

1. Query pre-processing module
2. Semantic Query formulation module

3.1 Query Pre-Processing Module

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The first step of the automatic proposed semantic query formulation approach is query pre-processing module. This module involved tokenization, stop words removal, and lemmatization. The output of the lemmatization used in part-of-speech tagger to assign a part-of-speech for each the query token for the further process of the queries. For example given a natural language query in figure 3.

Many prophets were reported to have been sent by God to the world according to Islam. Who is the last prophet and how can you prove that?

Figure 3. Natural Language query

Natural language queries are posed using different forms of words, so there is a need to reduce the inflectional forms of a word to a common base form. This section lemmatized the query tokens parsed by the system after stop words removal. This research used Stanford's CoreNLP Java library for lemmatization. After lemmatization, the imput query is then parse to part of speech tagger and assigns parts of speech to each word token, such as adjectives, nouns, prepositions, or verbs..

The tagged query tokens are used as input for the next module which is the semantic query formulation module. Figure 4 shows example of the pre-processed natural language query .

Many /prophets/ were /reported/ to/ have/ been sent/ by/ God /to /World /according/ to/ Islam. is /the/ last/ prophet/ can /prove /

Figure 4. Pre-processed Natural Language Query

3.2 Semantic Query Formulation Module

This module uses the pre-processed natural language query and transformed the query into structured triple representation. The module involves automatic concepts identification and automatic predicates identification with automatic disambiguation process and system-based suggestion in the instance when system fails.

3.2.1 Concept Identification

The main objective of the concept identification process tends to focus on automatically identifying concepts from the user query. Concepts identification involved parsing and matching noun query tokens against the gazetteer. Gazetteer is the list of objects or concepts in the knowledgebase.

The first task of concept identification is selecting noun query and identify such tokens as potential concept tokens. These potential concepts token are matched against the Gazetteer list. Therefore if any of the potential concept tokens are found in the gazetteer list, such token is automatically identified as a concept. That is to say, selected query tokens that are tagged as nouns (NN) or noun phrases (NP), are identified as *potential concepts or not.* For example, the system would remove /NN from the token *Islam/NN*, and *Islam* would be identified as a potential concept and parsed to the concept identification process. These *potential concepts* are then matched against the gazetteer list. After concept identification process, the system automatically identified *God* and *Islam* as concepts found in the user query as seen in figure 5.

prophet,report,God,Earth, Islam

Figure 5: Identified concepts

Figure 5 show example of identified concept from the user query. However, Due to the complex nature of natural language, problems may occur during automatic concept identification, such as:

- Ambiguity
- Constraint of vocabulary in Quran domain

For ambiguity problem in this paper, disambiguation process using automatic query expansion is employed. The experiment used the lexical dictionary WordNet to expand user query tokens with their respective synonyms. For Words that are not found in WordNet we use equivalent assertion for disambiguation. Equivalent assertion is the process of asserting that a particular word or concept is equivalent as a particular word. This technique is used to solve ambiguity for words that are not found in WordNet. It can be seen in figure 5, after automatic disambiguation of ambiguity. It can be noticed that the word *Earth* was included in figure 5 even though it wasn't in the user query. This is because the word *world* was used in the query in figure 3, but what was actually found in the gazetteer list is *earth*. So the system automatically disambiguates such word using the synonyms and identified it as concept.

After automatic concept identification, the next thing is to identify relationship between the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

concepts in order to form triple representation of the natural language query.

3.2.2 Predicate detection

Predicate detection is the process of identifying possible relations between the identified concepts or identified concept and any other concept in the knowledgebase.

The approach for predicate detection in this research employs a supervised statistical machine learning approach to detect possible relationship between the identified concepts by learning from the training set. In our approach, the training set is generated every time a query is posed to the system. That is to say, for every given query, the system automatically generates a training set based on the concepts that are identified in the query. For example, from Figure 5 *Prophet, God, Earth, Islam* were automatically identified as concepts, therefore the training set for these identified concepts will be all the triples in the triple store that have predicates relating the identified concepts. The predicate for the triples that are pulled are used as a training set for detecting possible predicates between the identified concepts. For example when the system attempt to identify possible predicate for the identified concept *Prophet*, the system was able to automatically generate the training set because *Prophet* matched objects of the triples *Isah, is-a, Prophet* and *Muhammad, is The Last, Prophet* from the triple store.

Given a training set containing predicates from generated triples, the system uses such predicates to learn, and identify predicates between identified concepts using the concept of N-gram maximum likelihood language modelling. The general formula for maximum likelihood estimation bi-gram probability is represented in Equation 3.3.

$$P(w_i/w_{i-1}) = \frac{count(w_{i-1}w_i)}{count(w_{i-1})} \quad \ldots\ldots\ldots\ldots 1$$

Equation 1 shows the maximum likelihood estimation for b-gram probability. The probability $P()$ that a particular word token or word W_n will precede a token or word W_{n-1} is the probability of their bi-gram. Here the system takes the co-occurrence of two tokens W_n and W_{n-1} and divides by the probability of the proceeding token or word W_{n-1}. So computing the maximum likelihood estimate of bi-gram probability $P(w_i/w_{i-1})$, i.e. the probability that the next word is W_n given the previous word W_{n-1}, is computed by dividing the number of times W_n and W_{n-1} occur together by the number of times the word W_{n-1} occurs in the training data. This will enable the computation to be normalized to the range between 0 and 1. The possible bi-gram with highest weight i.e. W_n, that proceeds W_{n-1}, is predicted to be the next possible word given the previous word. So in order to obtain the maximum likelihood of a phrase or sentence, the counts of estimated bi-grams are multiplied to obtain the most likely phrase or sentence given a training set. The probability of the maximum likelihood estimation for a phrase or sentence could also be represented in the form of log probabilities by adding the counts of the estimated bi-grams instead of multiplying.

The system begins predicate detection by estimating the maximum likelihood of unigram probability. Obtaining the unigram gives a stepping stone to the most likely word that could be used as the first word of the predicate we want to detect. For example computing the first unigram for the natural language query in figure 3 to detect the possible predicate for the identified concept Prophet. The system attempt to predict the first uni-gram by learning from the training set and estimate the probability that any of the query tokens in predicate lexicon could be the starting word of the generated training set for the identified concepts. The system iterates all the query tokens in the predicate lexicon and attempt detect predicate using probability of a word given the previous word. The first word in the query is *Many*. Therefore the system estimates the probability that the many is the start word can be of any predicate between the identified concepts:

$$P(Many/<s>) = Count\left(\frac{Many, <s>}{<s>}\right)$$

$$P(Many/<s>) = \frac{0}{2} = 0.$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$P(Many/is) = \frac{count(is, Many)}{count(is)}$$

Figure 7: Unigram Estimation

Figure 7 shows the first attempt to detect uni-gram with result being zero, meaning the words *many* cannot be the start word. Because none of the predicates in the training set has many as the starting word.

The systems automatically pick the next word in the query which is the word 'is' and do the estimation again. Here the system automatically identify that the word 'is' was used as the stating word of predicates in the training set. The same iteration is performed against each of the query tokens and the one with highest score is automatically predicted as out unigram.

After the system has automatically predicted the unigram, the next thing is to estimate the first bi-gram in other to predict the next word after *is,* i.e. $P(w_n/is)$.The first bi-gram will give us the likelihood of two words in the predicate occurring together to form a possible predicate for the identified concept, based on the frequency count in the training set. Since a predicate may be a phrase or sentence, it is likely that the first bi-gram will be a valid predicate. Therefore when the first bi-gram is successfully predicted, the system checks whether the bi-gram is a valid predicate. Here if the first estimated bi-gram is a valid predicate the system automatically detects such a bi-gram as the detected predicate, or else the system will make further attempt of estimating the next bi-gram given the currently predicted word.

For automatic estimation of the first bi-gram, the system iterates the predicate lexicon again and estimates the likelihood of bi-gram probability given the word *is*. The system therefore automatically starts again by taking the first word in the predicate lexicon, i.e. *Many,* and estimating the bi-gram probability given the previous word *is,* $P(Many/is)$ as seen in figure 8.

$$P(Many/is) = \frac{0}{2} = 0.$$

Figure 8: Bi-gram Estimation

Figure 8 Shows the estimation for the first bi-gram probability. The result of probability that the word *many* was followed by the word is *is* zero. The system therefore attempts the next word in the predicate lexicon, which is the word *were* and continue the process until bi-gram is estimated.

After several iterations, the words *is* happens to be followed by the word *the* in the training set, whereas *is* occurs twice, thereafter computing the probability as 0.5. Since it happens that $P(The/is)$ has the highest probability, the system automatically predicts that *the* is more likely to be the next word after *is*. Since the first bi-gram is estimated, the system automatically checks whether the estimated bi-gram is a valid predicate. If the bi-gram matches any predicate in the training set exactly, the system automatically concludes that the bi-gram is the detected predicate, or else the system goes further by attempting to estimate the second bi-gram. In the example query we are processing in this paper, the bi-gram '*is the'* did not form any valid predicate and thus the system goes further and attempts to estimate the next bi-gram. Therefore the system automatically reserves the word *is*, and uses the current predicted word *the* to predict the next likely word, i.e. To detect the next possible bi-gram. The system goes through the same process with the estimated first bi-gram probability $P(w/the)$, i.e. to predict whether any word in the predicate lexicon will follow the word *the*.

After several estimation, the system predicated first bi-gram is, '*is the'* with '*the'* having the highest count 0.5 and the second bi-gram predicts, '*the last*' word with a 0.5 score, and the third bi-gram is 0.05. Thereby predicting a possible predicate as 'is the last" now the system automatically checks whether the predicted sequence of words '*is the last'* is a valid predicate, and in this case the estimated words match exactly with the predicate '*is the last'* in the training set and thus the system automatically concludes that the predicate has been detected which is 'is the last'. The system therefore parses the detected predicate for the triple generation section.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

However, issues of concern may arise in automatic detection of predicates from complex natural language, queries, and they are:

- Ambiguity in predicate detection
- System may fail to automatically detect predicate

For dealing with the problem of ambiguity, during the process of automatic predicates detection, synonym detection is performed to estimate the maximum likelihood of a word or its synonym being given as the previous word. From our previous example, where the system attempted to estimate $P(wn/the)$, if a user uses the word *end* and this is represented as *last* in the knowledgebase, the system automatically resolves the ambiguity by predicting *last*, as *end* is used as a synonym of the word *last*. This enables an estimation of the maximum likelihood of predicting a phrase or sentence that relates to the identified concept semantically.

In the case of system failing to automatically detect predicate, system fail to detect possible predicate because of two season. First, the system may fail to automatically detect predicates due to a lack of identified concepts in the query token, i.e. the query tokens do not contain any concepts. Since the system automatically detects predicates based on query tokens that are identified as concepts, the system will fail to automatically detect predicates where there are no concepts in the query. For example, in *Can we marry more than one wife?* none of the query tokens is an ontology concept and thus, no concept will be identified. Secondly, concepts may be identified from query tokens but the remaining query tokens won't have enough information to detect any possible predicates. This may be due to a lack of corresponding information about the query in the knowledgebase: the concept used in the query was not annotated with any predicate that can be detected from the query token. For example, in *Is true that Ka'aba is the centre of the world?*the system will automatically identify *ka'aba* and *Earth* as concepts, but the remaining query tokens cannot be used to detect any valid predicate

The research proposed two the user two options. The user if presented with options to either reformulate his query or get suggestion from the system. If the user chooses to reformulate, the system allow user to reformulate the query words in order to re-process the query. If user choses to get suggestion from the system, the system based suggestion is presented to the user.

The system automatically uses the tokens to compute any possible predicate from the triple store that can be formulated based on query tokens. For example "Who is a generous person?" has no ontology concepts in the query token. However, there is a triple in the knowledgebase (Muhammad isAGenerous Messenger), and so the system will be able to compute the predicate "isAGenerous" from the query token. In this case the triple (Muhammad isAgenerous Messenger) is presented to the user as a suggestion. When the user is satisfied with a suggestion, the system automatically parses to the retrieval module for further processing.

Where the system is able to automatically identify concepts but fails to automatically detect likely predicates between the identified concepts, in our approach it automatically pulls out all triples from the triple store that involve the identified concepts and presents them to the user. For example, if the system is able to identify *Quran* and *God* which are both ontology concepts, but fails to identify any predicate from the user query token, it will automatically pull out all triples that are either (Quran, any relation, God) or (God, any relation, Quran) or (Quran, literal) or (God, literals) and present them to the user as suggestions. Here the triple chosen by the user is used as the triple for SPARQL query generation.

3.2.3 Triple Generation

The triple formulation process is the merging of the automatically identified concepts with the detected predicate to form triple. For example, from the automatically detected predicate in the example in Section 3.1, where *is the last* was detected, the system automatically forms a triple by merging the predicate with identified concepts that are related via such predicate. For example in this case, a concept in the knowledgebase is related to *prophet* via the predicate *is the last* as seen in Figure 9.

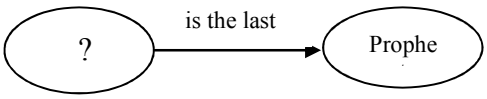

Figure 9. Triple generation

However, several possible triples may be formulated when the system is able to detect potential predicates between the identified concepts. In order to be more precise and return a closer triple representation of the user's natural language query, triple ranking mechanism is required.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.3Triple ranking

The ranking method is based on using Levenshtien string matching algorithm, a reverse engineering approach. In this method automatically formulated triples are matched against the triples in triple stored. The triple with highest score are presented on top. For example from the query in Figure 3, the system automatically generates two triples *?, is the last, Prophet* and *?, is-a, Prophet*. The system therefore automatically parses the generated triples to triple ranking section in order to get the most appropriate triple representation of the query. The triple ranking process starts by comparing the generated triple with those in the triple store, using a Levenshtien string matching algorithm training set. The ranking returned,*?,is-a, Prophet* has a higher weight than *?,is the last, Prophet*. However, *is-a, Prophet* is not the most appropriate triple representation of the query in Figure 3 because we are comparing it with any of the triples in the training set. Let's assume that the triples in the training set are *Muhammad, is the last, Prophet* and *Isah, is-a, Prophet* and we are comparing with these with the formulated triples. Because *Muhammad* has more characters than *Isah*, *?,is-a, Prophet* will have highest score than *?,is the last, Prophet*. However, although,?,*is-a, Prophet* has a higher weight it is most likely that *?,is the last, Prophet* is closer to a triple representation of what the user is trying to search for. Therefore in order to obtain the triple that is closer to the query words, this paper employed a reverse engineering approach by computing the distance between the ranked triples against the query, i.e. the minimum distance in transforming any of the ranked triples to the original query. This approach gave better results in terms of obtaining the triple representation most appropriate to the triple representation with the highest score. For example, after applying the reverse engineering approach, *?,is the last, Prophet* has a higher score and thus is accepted by the system as the semantically formulated triple. After ranking the triples, the top ranked triple is used is parsing to the retrieval module for retrieval of the relevant Quran verse.

In summary, this section has shown the step-by-step methods involved in the automated semantic query formulation approach based Statistical Machine Learning Technique. The chapter has provided comprehensive details of the proposed approach with detailed examples.

4. EXPERIMENT

For experiment Leeds University Quran ontology was used. The Quran ontology was annotated and stored in the Protégée ontology editor, which serves as a knowledgebase that responds to the semantically formulated queries. The statistics shows that a total of 300 nouns, i.e. noun concepts, obtained from Leeds Quran ontology were used. The number of predicate used for the experiment contained 350 relationship obtained from Leeds Quran ontology and additional relationship added during the development of Quran knowledgebase used for this research. A total of 82 queries obtained from the Islamic Research Foundation website were used for the experiment. The queries comprise 50 complex queries and 32 simple queries. Evaluation of automatic semantic query formulation approach is done based on the correctness of the semantically formulated and effectiveness of returned result by our approach compared to FREyA. The correctly formulated queries are measured based on formulated queries that returned relevant result using precision and recall information retrieval evaluation matrix.

Table 1 show examples of the queries uses for the experiment and automated triple generated for the natural language queries.

5. RESULTS

The results of Statistical machine learning approach for semantic query formulation in this paper will be compared with traditional keyword-based Quran retrieval, and the current systems which attempt to solve problems associated with the semantic query formulation task. A comprehensive evaluation was carried out to compare the proposed approach with the results provided by Quran domain experts, popular recent research in Quran retrieval Qurany, and the recent existing semantic query formulation approach, FREyA.

The comparison between the proposed semantic query formulation approach in this research and approach in FREyA was performed in terms of the number of queries that were semantically formulated correctly, the ability to correctly disambiguate ambiguous queries without failing, and the effectiveness of the system based suggestions provided by the both approaches.

Table 1. Example Of Triple Representations Of Natural Language Queries

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

No.	Query	Triple representation
1	Who is the father of Abraham?	(?, is the father of, Abraham)
2	Who is the mother of Jesus?	(?,is the mother of ,Jesus)
3	So many prophets have been reported to have been sent by God to the world, who is the last prophet among them and how can you prove that?	(?, is the last, Prophet)
4	How do we know that there is life after death? Please quote from a Hadith or the Quran.	(Life_After_death, is mention in, Quran)
5	I have heard so many stories about the people of Thamud, who was the prophet that was sent to the people of Thamud?	(?, send to people of , Thamud)

5.1 Evaluation of the Number of Correctly Formulated Queries

We evaluated the correctness of the semantically formulated queries by measuring the percentage of the queries that were semantically formulated correctly.

Table 2 shows that the average performance of the system in this study for the correctly formulated queries, which comprises both complex and simple queries, is 90.88%. 7.13% of both complex and simple queries were not formulated due to lack of corresponding knowledge in the knowledgebase. 2% of the queries failed. Phrase matching technique in FREyA, the average performance of the correctly formulated queries, which comprises both complex and simple queries is 73.5%. 7.13%% of both complex and simple queries were not formulated due to a lack of corresponding knowledge in the knowledgebase, and 19.38% of the queries failed

Characteristic Statistics		Total	(%)
No. of queries		82	100
Statistical machine learning using N-gram maximum likelihood estimation in this paper			
	Correct	74	90.88
	No Answer	4	7.13
	Fail	2	2%
Phrase matching in FREyA			
	Correct	60	73.5
	No Answer	4	7.13
	Fail	16	19.38

Table 2: Overall analysis of the correctness of the semantically formulated queries

Figure 10 shows the results of the performance of the proposed approach in this paper, in terms of semantically formulating complex natural language queries, in a graphical pie chart representation.

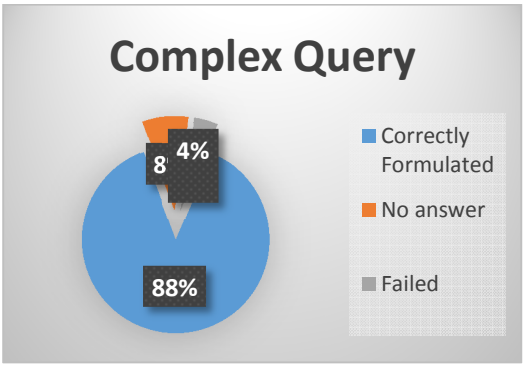

Figure 10: Complex query performance of proposed Automated Semantic query formulation based statistical machine learning approach.

Figure 11 shows the results of the performance of phrase matching technique in FREyA in terms of semantically formulating complex natural language queries in a graphical pie chart representation.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

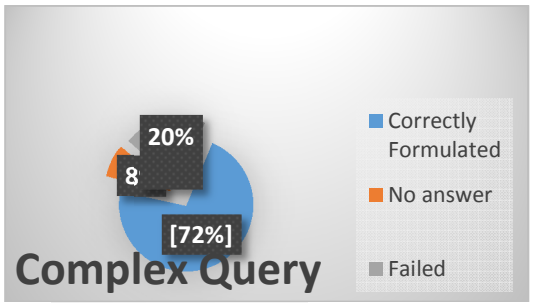

Figure 11: Complex query performance for phrase matching in FREyA

Figure 12 shows the results of the performance of the proposed approach in this paper, in terms of semantically formulating simple natural language queries, in a graphical pie chart representation.

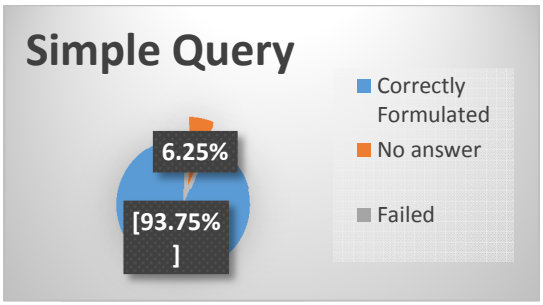

Figure 12. Simple query performance proposed Automated Semantic query formulation based statistical machine learning approach.

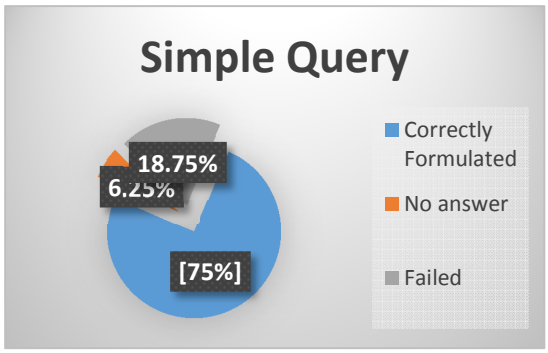

Figure 13. Simple query performance for phrase matching in FREyA

Figure 13 show the results of the performance of the proposed approach in this paper in terms of semantically formulating simple natural language queries.

Table 3: Analysis of the effectiveness of the suggestion approach

System Name	Precision	Recall
AutoSQuR	0.61	0.69
FREyA	0.58	0.62

Table 3 presents the evaluation of the suggestion approach in the proposed approach in comparison with the approach in FREyA. The evaluation of the suggestion approach is based on precision recall of the returned retrieved verses after suggestions were presented to the user and the user made a choice for further processing. The results show that the suggestions proposed in this paper had a precision of 0.61 and recall of 0.69 in terms of the Quran verses retrieved, while the suggestion approach in FREyA had a precision of 0.58 and recall 0.62 in terms of the retrieved verses after suggestions were provided to the user. The proposed approach of system-based suggestions in AutoSQuR outperformed that of FREyA in terms of precision and recall. This can be argued to the fact that in AutoSQuR Approach suggestion are provided automatically by the system based user's query. In the AutoSQuR approach the predicted and ranked triples are presented to the user in order to select for further processing. This allows the user to choose the best set of (subject, predicate, object) at the same time. In FREyA the disambiguation process is performed one concept at a time. Users are presented with suggestions in order to disambiguate queries one concept at a time. When a user disambiguates the concepts, they have to manually map the concepts with the respective suggested predicates. This process is time consuming, and if a user gets excited by the list of suggestions presented they may end up mapping concepts with suggested predicates that are not actually the right triple representation of the given query. As a result irrelevant Quran verses may be returned.

6. ANALYSES OF THE RESULT

The overall analyses shows the performance measurement of the proposed semantic query formulation based on statistical machine learning technique has outperformed the existing phrase matching technique in FREyA in terms of its ability to correctly semantically formulate simple and complex natural language queries to structured

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

triple representation by 17.4%. We argue that, the adoption of the statistical machine learning technique for the semantic query formulation with disambiguation approach and system-based suggestion has contributed to the improvement in the correctness of the semantically formulated query.

In summary, based on the analysis, what makes the research in this in paper better than the recent approach in FREyA is the effectiveness of the semantically formulated queries, which shows that the approach employed in this paper is better than the approach employed by FREyA. The effectiveness disambiguation process in this research was also better than the verses returned by Qurany and FREyA. This system also reduced user participation compared to the semantic query formulation process proposed in FREyA, where the user is involved in the disambiguation procedure. In this paper, the system performed disambiguation automatically. The suggestions provided by AutoSQuR when the system failed to automatically formulate natural language queries semantically, proved to be more effective and flexible than the approach in FREyA. The suggestion assisted users in suggesting triples for further processing, compared with the suggestion approach in FREyA where users chose to form the triples.

7. SUMMARY AND FUTURE WORK

In this paper, automated semantic query formulation approach that is able to formulated paragraph natural language query was proposed based using Machine learning approach. The approach used N-gram maximum likelihood estimation for the automatic approach. The paper improved from the current approaches of semantic query formulation approaches. Additionally, the approach in this paper proposed a solution to semantic query formulation approaches that failed to semantically formulates query due to lack of concept in the natural language query. Our approach proposed solution for this problem by prompting dialog to the user and ask the user if he intends to reformulate the query or needs system-based suggestion. If user chooses to reformulate, the system enable query reformulation. And if user chooses to get suggestion from the system, system based suggestion is provided to the user based on concept matching and N-gram maximum likelihood estimation. This solution provided by the system assisted in reducing the number of failed queries and thus increase effectiveness of the system.

For experiment, Quran ontology form Leed University, UK was used as a test-bed to semantically formulate natural language queries. The idea of semantically translating natural language queries to structured queries involves the transformation of natural queries of any length to the same structure of data representation in the knowledge base. Here, natural language query is automatically formulated into formal structured triple representation *(subject, Predicate, Object)* or *(concept, Predicate, concept)* to enables the retrieval of semantically structured data in RDF triple format of the knowledge base. The automated semantic module was implemented using statistical machine learning technique to automatically generate triple representation of the natural language query based on examples in the knowledge base based on N-gram maximum likelihood estimation.

Furthermore, in order to increase effeteness of the semantic query formulation, when more than one triple is generated by the system, triple ranking was presented based on Levenshtien string matching algorithm a reverse engineering approach. The triple ranking enables automatic processing of the most likely triple representation of the query. Ranking triple has proven more effective in terms of processing the closer triple representation of the natural language query than predicate ranking. When the system automatically formulates triple from the user's natural language query.

Future work will be incorporating the Hadith ontology by merging Quran and Hadith ontology into the semantic search in order to improve the effectiveness of the semantic query formulation process and the returned results.

Another Part of the future research challenge that is not addressed in this paper is Boolean queries. The system in this paper doesn't cover yes/ no or true/false questions which are quite popular in Islamic related queries. In our future work, we intend to incorporate support for yes/no queries so that users can ask queries to which the target answer is yes or no, or true or false.

ACKNOWLEDGEMENT

The authors would like to delicate their sincere thanks to Universiti Kebangsaan Malaysia (UKM) for funded this research under research grant code GGPM-2015-003.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] Bah, A., & Carterette, B. Aggregating results from multiple related queries to improve web search over sessions. In *Asia Information Retrieval Symposium. 2014* pp. 172-183. Springer International Publishing.

[2] Ameen, A., Khan, K. U. R., & Rani, B. P. Reasoning in Semantic Web Using Jena. *Computer Engineering and Intelligent Systems*, 2014. *5*(4), 39-47.

[3] Solskinnsbakk, G. ,Contextual, "Semantic Search Navigation", *PhD Thesis*, Norwegian University of Science and Technology, 2012.

[4] Tablan, V., Damljanovic, D., Bontcheva, K., "A natural language query interface to structured information". *In: Proceedings of the 5h European Semantic Web Con-ference (ESWC 2008). Lecture Notes in Computer Science, vol. 5021, pp. 361{375.Springer-Verlag New York Inc,* Tenerife, Spain ,June 2008.

[5] Damljanovi, D. D. , "Natural Language Interfaces to Conceptual Models", *PhD thesis*, University of Sheffield, UK, 2011.

[6] Ahmed, Z., & Gerhard, D.,"Web to Semantic Web & Role of Ontology", *In the proceedings of National Conference on Information and Communication Technologies*, (NCICT-2007), Pakistan, 9th May 2007.

[7] Machado, C. M., Rebholz-Schuhmann, D., Freitas, A. T., & Couto, F. M. The semantic web in translational medicine: current applications and future directions. *Briefings in bioinformatics*, 2015. *16*(1), 89-103.

[8] Tauberer, J., "Why we need a new standard for the Semantic Web", (October 2005). Retrieved from http://www.rdfabout.com/intro/?section=1

[9] Hushon, John Daniel. (2014). "Context-driven model transformation for query processing." U.S. Patent No. 8,812,452. 19 Aug.

[10] Kassim, J.M., Rahmany, M. ,"Introduction to semantic search engine". *In: International Conference on Electrical Engineering and Informatics*, 201). 380–386. Selangor, Malaysia

[12] Kharlamov, E., Giese, M., Soylu, A., Zheleznyakov, D., Bagosi, T., Console, M.,Waaler, A. "Semantic Access to Big Data the Case of Norwegian Petroleum Directorate's Fact Pages". *The 12th International Semantic Web Conference*, 21-25 October 2013, Sydney, Australia (1), 1–4.

[13] Wimalasuriya, D. C., & Dou, D. Ontology-based information extraction: An introduction and a survey of current approaches. 2010.

[14] Aygul, F. A., Cicekli, N. K., & Cicekli, I. Natural Language Query Processing in Multimedia Ontologies. In *KEOD* 2012. pp. 66-75).

[15] Cordier, A., Gaillard, E., & Nauer, E. Man-machine collaboration to acquire cooking adaptation knowledge for the TAAABLE case-based reasoning system. In *Proceedings of the 21st International Conference on World Wide Web* 2012. pp. 1113-1120). ACM.

[16] Franconi, E., Guagliardo, P., Tessaris, S., & Trevisan, M. A natural language ontology-driven query interface. In *9th International Conference on Terminology and Artificial Intelligence* 2011. pp. 43

[17] Barzdins, G., Liepins, E., Veilande, M., &Zviedris, M. "Ontology Enabled Graphical Database Query Tool for End-Users".*In Eighth International Baltic Conference on Databases and Information Systems (DB&IS* 2008), 105–116.

[18] Guha, R., McCool, R., and Miller, E. "Semantic Search". *Proceedings of the WWW2003*, Budapest, 2003.

[19] Gauch, S., Chaffee, J., Pretschner, A., "Ontology-Based User Profiles for Search and Browsing". *Web Intelligence and Agent Systems* 2003. 1(3-4), 219–234

[20] Akula, A. R. *A Novel Approach Towards Building a Generic, Portable and Contextual NLIDB System* (Doctoral dissertation, International Institute of Information Technology Hyderabad). library". *Data & Knowledge Engineering*, 2015. 55(1), 4–19.

[21] Cimiano, P., Haase, P., Heizmann, J., Mantel, M., &Studer, R. "Towards portable natural language interfaces to knowledge bases – The case of the ORAKEL" system. *Data & Knowledge Engineering*, 2008. 65(2), 325–354,.

[22] M. Damova, D. Dannelles, R. Enache, M. Mateva, and A. Ranta. "Natural language interaction with semantic web knowledge bases and lod. In Towards the Multilingual Semantic Web". *Springer*, 2013.

[23] R. Shobana, D.Venkatesan "FFQI-Fast Formulation Query Interface For". *Journal of Theoretical and Applied Information Technology* 2012. 37(1), 125–131,.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[24] Lopez, V., Motta, E., Uren, V. and Pasin, M.,"AquaLog: An ontology-driven Question Answering System for Semantic intranets". *Journal of Web Semantics.* 2007. 5 (2),.

[25] Kaufmann, E., Bernstein, A., & Zumstein, R. (2006, November). Querix: A natural language interface to query ontologies based on clarification dialogs. In *5th International Semantic Web Conference (ISWC 2006)* (pp. 980-981). Springer.

[26] Tablan, V., Damljanovic, D., &Bontcheva, K. "A Natural Language Query Interface to Structured Information", *In ESWC 2010, volume* 6088 of LNCS,

[27] Lopez, V., Fernández, M., Stieler, N., Motta, E., Hall, W., Mkaa, M. K., & Kingdom, U. "PowerAqua : supporting users in querying and exploring the Semantic Web content", *Semantic Web Journal,*2011.

[28] Lehmann, J., & Lorenz, B. ,"AutoSPARQL : Let Users Query Your Knowledge Base". *In Proceedings of ESWC* 1–15, 2011.

[29] Yahya, M., Berberich, K., &Elbassuoni, S., "Natural Language Questions for the Web of Data". *In Proceedings of the 2012 joint conference for Empirical methods of Natural Language Processing and Computational Natural Language Learning,2012*.

[30] Nurfadhlina Mohd Sharef, And ShahrulAzman Noah. , "Natural Language Query Translation For Semantic Search". *International Journal Of Digital Content Technology And Its Application,*2013.

[31] Damljanovic, D.; Agatonovic, M.; and Cunningham, H. "FREyA: an Interactive Way of Querying Linked Data using Natural Language. *The SemanticWeb*. 2011. Pp. 125–138. Springer,.

[32] Habernal, Ivan, Konopík, Miloslav,"Expert Systems with Applications SWSNL : Semantic Web Search Using Natural Language". *Expert Systems with Applications*. Volume 40, Issue 9, 2013, pp. 3649–3664,.