Review on Semantic Query Formulation Approaches

Rufai AliyuYauri, Haruna A Yelwa
Information and C Communication Technology Department
Kebbi State University of Science and Technology, Nigeria.
Multi Media Uinversit, Malaysia
*rufaialeey@yahoo.com , yauriharunaaliyu@yahoo.*com

Abstract-Semantic Web plays important role in the access of Web data based on the meaning instead of traditional keyword search. However, the main challenges of semantic search systems is complex structured queries such as SPARQL are needed in order to retrieve data semantically. To address this challenge, several efforts have been put in place by researchers to enable user use natural language query for retrieval instead of using complex structured query syntax. Natural language query is transformed into structured query for retrieving semantically which is refers to as semantic query formulation. This study presents a review on semantic query formulation researches that has been reported by researchers. The main motivation of the paper is because of the growing interest semantic search systems globally. The study will give an up to date works on the area of semantic query formulation which will assist in furthering up researches in the area

Keywords: Semantic Web, Ontology, Semantic Query Formulation, Retrieval

1.0 INTRODUCTION

The semantic web is described as a web of linked data. (Shadbolt, Berners-Lee & Hall 2006) define the semantic web as “An extension of the current version of web where information is given well-defined meaning, enabling computers and people to work in co-corporation”. It is designed to overcome the challenges of the current web search systems, which are mainly designed for presenting, organising and linking data in the form of text, videos, audio and images. The structure of data on the current World Wide Web is published in such a way that the data can only be understood by humans, computer programs cannot understand its meaning. The web search systems struggle with the aggregation and querying of information without having a consistent way of achieving such tasks, whereas the semantic web concept enables linked documents on the web and assigning better meaning for both human and computer understanding. In other words it improves the current world wide web structure from that of interconnected documents to semantically driven documents that allow better aggregation of information, storage, manipulation and retrieval (Kück, 2004). This provides a better enabling environment for promoting good working relationships between humans and computers. The main building block of semantic web is ontology.

The main building block of the semantic web is ontology, which transforms web content into a machine-readable format that can be manipulated (Ahmed & Gerhard, 2007). Ontology is the main building block of the semantic web which transforms web content into a machine-readable and format that can be manipulated (Ahmed & Gerhard, 2007). 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 (Hu, 2004). In the semantic web, data is represented in formal ontology format. Where ontology model data is in the form of concepts and relationships between concepts. In the semantic web these concepts and their corresponding relationships are represented in RDF graphical format. RDF is World Wide Web Consortium’s (W3C) standard syntax for representing concepts and relationships between concepts in graphical format. RDF shows or describes the relationship between a concept and its literal. (Patel-Schneider, 2005). It is presented in triple as below.<Subject>< Predicate><Object>

Subject and object represents an ontology concept, while the predicate represents the relationship between these concepts. RDF triple format was mainly a practical rule language for computers to understand, manipulate and share data (Decker, Sintek, Andreas, Nicola, Andreas, Leicher, Susanne, Weathers , Gustaf &Zolt,2007.). Therefore, an RDF graph shows data represented in a format by which computers could understand the meaning and processes. RDF triple formats are stored in a knowledgebase, which enables computers to access the data and manipulate it semantically.

Semantic searches are seen as a semantic web technology approach to interpreting search queries and resources based on underlying ontologies, labelling some contextual domain knowledge, by connecting web resources to semantic annotations (Fazzinga & Lukasiewicz 2010). A semantic search is a data retrieval mechanism that integrates the capabilities of the semantic web and search engines in order to get more precise results than the current search engine. The main basic concept of a semantic search is that it semantically manipulates and transforms a natural language query to a structured formal query such Protocol and RDF Query Language (SPARQL), RDF Data Query Language (RDQL),Sesame RDF Query Language(SeRQL), Triple pattern matching among others (Tablan et al.,2008.; Esmaili & Abolhassani, 2006). These structured queries enable users to retrieve data from knowledgebase and other knowledge-related sources.

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). A structured query involves complex syntax and therefore users need to be familiar with this before it can be used. Figure 1 shows an example of a structured query in 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)

The query above is a structured query in SPARQL syntax, and in natural language means “select all names that are in friend of friend data (foaf). The user therefore needs to have prior knowledge of what they should query, which is too complex for users and requires them to learn the syntax before they can query the knowledgebase using structured formal query language. To simplify access to data in the knowledgebase, users should therefore be allowed to use their favourite natural language query, so that they can pose their query using a natural language and system which assists the user to formulate the query into a structured query which is what is refers to as semantic query formulation.

Semantic query formulation is the transformation of a natural language query into a formal structured query such as SQL, SPARQL, and SeRQL, among others. The goal of query formulation is to assist users by formulating their natural language query into a formal structured query in order to enable them retrieve relevant information from a knowledgebase.

In recent years, there has been an increase in research interest about the semantic web and semantic search engines, so as to enable users to have easy access to structured datasuch as RDF data representation in the knowledgebase. The semantic search approaches mentioned earlier have been adopted by different semantic query formulations systems. We will examine the various research that has reported on semantic query formulation in the next section.

2.0 PREVIOUS RESEARCH ON SEMANTIC QUERY FORMULATION

In recent years much more information has become available on the web, in databases, knowledgebase and other related document storage and manipulation mechanisms. Quite a number of organisations use automated information systems, mostly for storage of their information. Many important decisions are made based on adequate support for information, which remains a problem, as retrieval is based on complex schemata. This has led to a number of research projects that focus on the area of query formulation (Hofstede, Proper, & Nijmegen, 1995). Several research projects have been proposed that attempt to semantically search for knowledge stored in a knowledgebase. Currently, there are couple of semantic search research projects that are designed for retrieval of a knowledgebase from different domains. These semantic search systems attempt to semantically search knowledge represented in the knowledgebase by semantically formulating user queries in various distinct ways (Blanco, Halpin, Herzig, Mika, Pound, Thompson, Tran &Thanh, 2013; Madhu, Govardhan, & Rajinikanth, 2011). Previously reported systems on semantic query formulation can be categorised into three approaches: mainly manual, semi-automatic and automatic semantic query formulation systems. Most of these semantic query formulation approaches formulates natural language queries into triple format and then to corresponding SPARQL, because SPARQL is the most powerful formal language for retrieval from a knowledgebase, as recommended by the W3C group.

2.1 MANUAL SEMANTIC QUERY FORMULATION.

The manual semantic query formulation process is a semantic query formulation approach where a user semantically manually constructs a structured query language such as SPARQL. The user manually writes the syntax or represents their query in the same format as the knowledgebase data, such as triple. The manually constructed semantic query is then used against the knowledgebase for retrieval.

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 to manually formulate a formal query language and retrieve knowledge from the knowledgebase. Protégée enables users to manually construct a SPARQL query in order to retrieve knowledge stored in Protégée. In Protégée, a user can use the SPARQL query tab provided to construct a SPARQL query, and execute the query against the knowledge stored in Protégée. The result of the corresponding constructed query is a return in the form of triples or concepts to the user depending of what they are looking for.

Virtuoso SPARQL Query Editor is another system that enables users to construct SPARQL queries in order to retrieve information. Virtuoso presents an interface to the user where the user formulate the SPARQL query, and the system then formulate the SPARQL query to the equivalent SQL query to retrieve data from triple store tables. In Virtuoso, all graphs are stored in one large table containing annotated ontology concepts.

Another query editor that allows users to manually construct formal queries was presented in the work of (Kharlamov, Giese, Soylu, AZheleznyakov, Bagosi, Console, Haase, Horrocks, Marciuska, Pinkel, Ruzzi, Santarelli, Savo, Sengupta, Schmidt, Thorstensen, Trame & Waaler, 2013). The system is a visual semantic query formulation system that poses queries via a visual query formulation (VQF) interface. VQF is a SPARQL query editor that allows a user to manually construct a semantic query. The semantically formulated queries are then executed by the query answering module against a knowledgebase for retrieval of the answer. The process of manual construction of the semantic query is complex because it requires users to be familiar with the complex syntax of the query language before retrieval from the knowledgebase. Work in (Popov, Kiryakov, Kirilov, Manov&Dimitar, 2003; Damljanovi, 2011) improve from manually semantic query formulation to semi-automatic semantic query formulation were proposed. In these system computer and human work together in order to semantic formulate structured query.

2.2 Semi-Automatic Semantic Query Formulation

Unlike the manual semantic query formulation approach, where users are required to manually formulate their query without assistance from the system, in the semi-automatic semantic query approach, the system and user work together in the query formulation process. In this approach, the user is given some sort of assistance by the system in order to semantically formulate their query, which is used to retrieve knowledge from the knowledgebase. Most of the systems using the semantic query formulation approach are based on the semi-automatic semantic query formulation approach. The semi-automatic semantic query formulation approach comprises of a combination of the automatic and manual approach where computers and humans collaborate to semantically formulate a formal query. In this approach, users don’t need to be fully familiar with the complex syntax of a formal query language or know exactly how information is represented in the knowledge before they can pose their query and get answer.

Some of the semi-automatic semantic query formulation systems are template-based, where the user is presented with menu-based information from which they choose variables that are used for semantic query formulation. The template base query formulation approach is presented in the KIM system (Popov. et al, 2003) It was designed to go a step further than the manual semantic query formulation system. The KIM approach was mainly to combine automatic and manual semantic query formulation approach. The system saves the user from going through the complexity of a structured query and the effort of knowing the structure of the knowledgebase, before querying and retrieving a result. In this system, the user is presented with predefined query templates from where they choose to semantically formulate the structured query SeRQL that is used for retrieval from the knowledgebase. The SeRQL translation is then used to match data in the knowledgebase for retrieval.

Another semi-automatic approach for semantic query formulation is (Donderler, Saykol, Arslan, Ulusoy & Gudukbay 2003). Here information in the knowledgebase is presented to the user in graphical format where the user sketches to formulate semantic query. In this approach, a visual query is formulated by a gathering of objects with some conditions. Here the system provides the user with a visual interface which provides a graphical representation of the knowledge from which the user chooses variables for query formulation in order to retrieve important knowledge. Another work that focuses on query formulation in database was presented by (Chen & Zhu, 1998). The conceptual modelling approach allows users to express the way they intend to discover knowledge from a database on a constructed network. The user chooses variables that form a network, which provides the system with a hint of what the intended query should look like to form a casual network. If a causal network is formed, it could indicate possible relationships between some concepts. When the user send a query, the relevant stored knowledge is presented, from which they choose and guide the knowledge search. Another work where users select query variables for query formulation in a database is (Bellavia, Maio, & Rizzi, 1992)This provides a query formulation system which supports inference about problems by optimizing query formulation cost. The system is based on the concept of the user viewpoint (UV), where a user chooses their viewpoint by defining criteria for accessing data. The user selects a relation from the database schema. The query is formulated based on this relationship by using the semantic aspect of the selected relationships. The relationship chosen by the user is used to formulate a query, taking into consideration various graphical links between concepts according to the selected relationship in the database. (Dongilli, Franconi, & Tessaris, 2000) present another ontology based query processing system that semantically formulate a user's natural language into a structured query. This is a project by Semantic Webs and Agent Integrated Economies (SEWASIE). The project focuses on building an intelligent natural language query interface that supports users in formulating their natural language query into a precise query. The system is based on user/ computer interaction where the user is presented with a visual interface to query ontology. The system uses the user's query to provide various related concepts and relationships for the user to choose from in order to retrieve the desired information. This system also requires the user to be involved in an initial query formulation task, by selecting concepts that should be used for query formulation. Here the user is also restricted to words in the provided in the knowledgebase. The SEWASIE system supports users in formulating a semantic query based on the refinement process supported by ontology navigation. Users specify a query using generic terms, are able to refine some terms, can also introduce new terms and can iterate the process if required by the query.

Barzdins, Liepins, Veilande, & Zviedris (2008) present an ontology-assisted query formulation based on concepts annotated in the database. The main concept of this approach is to use ontology as the main guide to generating a SPARQL query. The user query is formulated with the help of an assisted graphical user interface that enables them to construct the semantic query. The user first chooses concepts in the ontology that are related to their query and relate with available relationship that connects the selected concept. The system uses the shortest path algorithm to find available relationships that exist between the selected concepts found in the ontology.

Although the above-mentioned systems simplified by the effort of semantic query formulation compared to the manual approach, however this system has some limitations because users are restricted to information presented to them by the system from which they choose in order to semantically formulate their query for retrieval of the desired information. In the template-based approach, the user needs to browse a lot of information provided by the system before they can formulate the query that is used for retrieving important information.

TAP (Guha, McCool & Miller, 2003)AmiGO(GeneInfoViz, 2007) and Gauch, Chaffee, & Pretschner (2003) proposed some improvements to the approaches mentioned above. TAP goes beyond just providing a template from which the user chooses the variables that are used for query formulation, by providing the users with a search and browse mechanism. 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 (Damljanovi, 2011). There is therefore an 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 when a user is using the word "Allah" but this is represented as "God" in the underlying knowledgebase.

CINDI improves further from the previous systems by looking at the problem of ambiguity in user queries. CINDI proposed a form-base query formulation approach where a user poses their query through filling in a form presented by the system. CINDI formulate user natural language queries to structured language SQL using semantic templates (Stratica, Kosseim, & Desai, 2005). The system provides the user with a template that enables them to formulate a structured SQL query. The query is syntactically parsed by Link Parser, and semantically analysed based on domain-specific templates. These templates are connected to a conceptual knowledgebase from a database schema using WordNet. The semantically stored information is used to guide the user in formulating an SQL query by selecting the concept and relationship. The system was tested on the CINDI database containing information about virtual libraries. In this system they incorporated lexical dictionary WordNet to create a list of hyponyms and synonyms for each relationship and attribute name. This enables the user to query the knowledgebase with flexible words without being restricted to vocabularies provided by the system.

QUICK (Zenz, Zhou, Minack, Siberski, & Nejdl, 2009) is a work on query formulation that supports users to construct structured queries.

Their system is based on user interaction, where the user is guided through constructing a structured query based on the underlying ontology backend. In this system the user makes the query, the system uses the semantically annotated information in the knowledgebase and key works in the user query to provide the user with a clarification dialogue where they choose their intended semantic query. This system also requires user participation in selecting query variables for the system to use in formulating the query. In this case the user queries the system, and the system processes and provides the user with a clarification dialogue, to which the user provides clarification for the intend query. The system then uses the clarification provided by the user in formulating the user's natural language query for information retrieval. This system also uses WordNet to enables flexible vocabulary without restriction, but the system is based on a keyword search, which restricts the keywords that can be used since a keyword may not necessarily be in existence even if the synonym of the word is checked.

LANLI is a natural language interface system for relational database query formulation (Enikuomehin & Okwufulueze, 2012). The work semantically assists user's natural language queries to retrieve information from databases. In this work, the system allows users to ask queries in natural language, and they are transformed to structured query language SQL. The SQL is executed over a relational database for retrieval of important information. In LANLI they proposed the use of an Unguided Loose Search where users can simply send their request in terms of natural language or present some set of the keywords that describe their desired information without worrying about the database structure or syntax. In this system WordNet is used to expand the user's words during query formulation.

Quite a number of works have been presented that offers users a Google-like search mechanism where users pose natural language queries to retrieve information from the knowledgebase. These systems accept user to query using natural language and the systems semantically assist the user in formulating the query as a formal structure query which is then used against the knowledgebase to retrieve relevant information. ORAKEL is a natural language interface system that semantically formulate a user's natural language query into a structured query (Cimiano, Haase, Heizmann, Mantel, & Studer, 2008).ORAKEL is a system that accepts user input as natural language queries to the system and the system formulates the queries into formal structured queries in order to be matched against the knowledgebase for retrieval. ORAKEL approach formulates factoid questions such as what, who, where, and which, using full syntax parsing and a compositional semantics approach. The user's natural language is processed and formulated according to the underlying knowledgebase.The system identify concepts from user query work and user to manually associate the concepts with relationships in the underlying knowledgebase.

The system is based on two fundamental roles comprising of the end user who uses the system by querying the system, and the domain experts who are familiar with the underlying knowledgebase and play the role of lexicon engineers who interact with the system in lexicon acquisition mode. The lexical engineer creates domain-specific lexicons to adapt the system to an exact domain. In this case, users ask questions which are semantically interpreted by the query. A user's query is formulated by taking into consideration the user's query where concepts are identified by the system and the user manually chooses the relationship with respect to domain-specific predicates. The system simplifies formulating structured query in order to retrieve from the knowledgebase.

PANTO is a natural language interface approach where the system accepts a natural language query and transforms it into a structured query (Wang, Xiong, Zhou, & Yu, 2007). The system accepts generic natural language queries, transforms the queries and output it inform of structured query language SPARQL. The system is a triple-based query formulation approach that is designed to cope with various natural language issues such as negation, superlatives and comparatives. It involves use of the statistical Stanford parser, WordNet, and various metric algorithms that transform natural language queries to triple-based representation. The triple representation enables the construction of a SPARQL query language based on user interaction.

Aksac, Ozturk, and Dogdu (2012) present PERSON, which is a system design to semantically search for relevant information from the web. The approach is to extend the functionality of the Mozilla firewall browser by incorporating a plug-in that identifies named entities navigated by users, annotates such entities and formulate the queries that retrieve relevant information based on associating newly annotated data with the existing data for retrieving relevant information. Users are guided to a query by clicking ontology entities, and then a SPARQL query is executed and related knowledge from the ontology is retrieved.

Damova and Dann (2013) 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. A user's natural language is transform into SPARQL in order to retrieve answer mainly yes/no answers. In this system the user needs to search knowledge base to get the best desired answer. There is no proper ranking mechanism to get the best-formulated query to be transformed into SPARQL, since the system returns many possible options.

Furthermore another interactive query formulation system that focuses on transforming a user's natural language query to a formal structured format was presented in (Jarrar et al., 2012). In this work, they propose an interactive query formulation language (MashQL) to enable users to access data on the web easily, with more precise answers to queries. The language is represented in the form of a tree, where the roots are seen as the query subject and each branch is comprised of properties and restrictions. The system allows users to navigate through and select various concepts and relationships from an underlying dataset. Users interact with the system to form various questions by selecting concept relationships and restrictions to form simple and complex queries.

FFQI presented a query formulation approach for retrieving structured data from database (Shobana and Venkatesan, 2012). 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.

SemSearchis a keyword-based search system that semantically transforms user keyword queries to formal structured queries (Lei, Uren, & Motta, 2006). The system accepts a natural language query and transform it into structured SQL query language. The system is based on a concept search where users send queries as concepts in order to get result based on that concept. In SemSearch the user conveys to the search engine the type of search result. In this case the user is required to have knowledge of the concepts that exist in the knowledgebase before they can query and retrieve important information. NaLIX is another semantic search system that is based on user interaction for retrieval from XML data (Li, Ave, & Arbor, 2005). The system accepts a natural language query and formulates the query into an XQuery expression, which is used against XML data for retrieving important information. The system maps the grammatical closeness of natural language parsed tokens to the closeness of corresponding elements in the result XML. In this system users interact with the system by selecting a template from which they choose from natural language queries that are already loaded, or users make a natural language query and the system guides the user with suggestions for how to make a suitable query that could be answered by the system.

ONLIstands for 'ontology natural language interaction' and is another natural language question answering system that is mainly designed to transform user's natural language queries into nRQL. The system is based on users being familiar with the ontology concept in order to transform their query into a structured query. User interaction is needed for query transformation to nRQL. Unger et al., (2012) describe a template-based question answering system for querying RDF graphs. The system's main purpose is to assist users in querying RDF graphs by transforming natural language user queries to SPARQL. It reflects the internal structure of the question using statistical entity identification and predicate detection. The system attempts to identify concepts in a user's query token and detect possible relationships that may exist between the identified concepts based on deep linguistic analysis by generating SPARQL templates with slots that need to be filled with URIs. To fill those slots, the likely concept is identified using string similarity and natural language patterns extracted from structured data and text documents.

QASYO is another system based on a question answering approach where a user's natural language query is transformed to a structured query (Moussa & Abdel-kader, 2011). The system accepts natural language queries and YAGO ontology as input and retrieves information from the semantically annotated data. The system analyses natural language queries by looking at the keywords in the query, mapping the keywords against the semantically annotated data, and retrieving answers relevant to the user query. The system is based on triple mapping, where user queries are transformed into triple form and matched against the triple represented data in the knowledgebase.

Although the semi-structured semantic query formulation system eases the processes involved for user to retrieve data from a knowledgebase using natural language, the time and the processes involved still require a lot from users. Users need to interact with the system before they can retrieve important knowledge from the knowledgebase, which is hectic and time consuming. Users need a more simple method such as a Google-like search mechanism where they can easily make a query and receive an answer without participating in the retrieval process.

2.3 Automatic Semantic Query formulation Approach

In order to fully automate the process of semantic query formulation, various systems attempt to automate the process of semantic query formulation. Although the proposed automatic semantic query formulation systems have shown significant improvement over the semi-automatic query formulation approach, most of these systems are still not fully automated. The systems still involve users during the semantic query formulation process. There is still not a fully automated system that automatically formulate a user's natural language query to a structured query without human intervention.

AquaLog(Garcia, Hall, Keynes, Motta, & Uren, 2006) presented a system based on a question answering approach. The system is a portable Natural language interface that enables users to query the knowledgebase using natural language. A user is able to make a query using natural language, the system analyses the query, formulate the query into a structured query and matches the query against the knowledgebase in order to make an inference and return an answer to the user. This system attempts to semantically formulate structures query from user's imputed natural language query. When a user send their query, the system starts by automatically transforming the user query into possible linguistic triples. Linguistic triples are candidate triples generated from a user query after some linguistic processing, which are used to generate the best triple representation of the user query through the Relation Similarity Service module (RSS). Aqualog is potable because it takes natural language queries and ontology as an input, and then returns answers retrieved from one or more knowledgebase. Users can ask queries and customise their queries by associating some keywords with the concept in the ontology.

NLP-Reduce is an approach based on automatic semantic query formulation that transforms a user's natural language query into a structured query (Kaufmann, Bernstein, & Fischer, 2007). The system presents a natural language interface that transforms natural language queries into structured queries. 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 able to enter keywords or full sentences for querying the knowledgebase**.** The system uses a set of natural language processing techniques, such as stemming and synonym expansion to reduce a query to a structured query.

PowerAqua(Lopez, Fernández, Motta, &Stieler, 2011) is an ontology-based natural language interface that supports the transformation of a user's natural language query into structured form. It is an extension of the previously discussed Aqualog, mainly designed to cope with problems in the Aqualog system. Power Aqua uses huge amounts of available heterogeneous semantic data in order to interpret a natural language query, without making any assumptions about the particular ontologies of a particular query. 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.

QACID is a semantic query formulation system for querying database in a natural language question answering approach (Ferrández, Izquierdo, Ferrández, & Vicedo, 2009). The system transforms collections of given domain queries, analyses such queries, and groups the queries in clusters. Each query is used for mapping natural language query terms with knowledge in the knowledgebase by using string distance metrics. Each cluster contains representations of a certain group of queries and has a characteristic query pattern derived from training set data.

AutoSPARQL is a query formulation interface that formulates user queries to SPARQL query language (Lehmann. et al, 2011). The system is based on a supervised machine learning approach that learns about a user's intended query based on user interaction. Users can either ask a question directly, as in a question answering system, search for relevant resources, or select a search result. Although the system claims to be an automatic SPARQL generation system it relies heavily on learning from what users describe as answer or not answers, i.e. positive and negative answers. The system first allows users to search for a concept, for example *animal*, then the system will ask the user if the answer to the search term is "Donkey" if they say yes, then it is used as a positive answer, and if they say no it is used as a negative answer. The system learns answers to certain queries, which can be used to show the next user asking a similar query.

Deines & Krechel (2013) describe another semantic query formulation approach that formulates a user's natural language query in German to SPARQL query language. The system uses natural language queries in the German language and matches them against an RDF graph that is labelled in German. The system is based on identification of various resources from user queries, which shows path-based identification of similar semantic resources. In this system users pose questions in the German language, the system then automatically expands the domain ontology used by populating the ontology with the corresponding German vocabulary using GermanNet.

DENNA is another semantic query formulation system mainly designed to transform natural language queries into structured queries (Yahya, Berberich, & Elbassuoni, 2011). The system is based on integer linear programming for solving several natural disambiguation problems. The system transforms a user's natural language query into a structured SPARQL query, where the focus is on entities, classes, and the relationships between them. The system utilizes the richness of the knowledge base's semantically annotated data which enables users to make a query in natural languagThe MyAutoSPARQL system (Sharef, Noah, 2003) 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. MyAutoSPARQL started by using linguistic processing to transform user queries into linguistic triples, as in Aqualog. The system then tries to identify the namespace of the concepts and obtain the closest property name based on the linguistic triples, and then the triples are formed for SPARQL construction. Although the system attempts to automatically transform natural language queries into structured SPARQL queries, it possesses some limitations. First the system relies on linguistic triples formulated by the system in order to process to SPARQL construction, which is based on heuristic linguistic processing.

Systems were developed in order to deal with this problem by attempting to automatically formulate a user's natural language query to formal query language but, if the system is not able to semantically formulate user queries, they don't just fail. These systems try to assist users to avoid having to re-write their queries from the scratch. This gives the user some sort of support in case their queries cannot be answered directly by the system. The Querix system employs semantic query formulation of users' natural language queries (Kaufmann, Bernstein, & Zumstein, 2006). The system is in the form of a natural language interface, where a user is able to make queries using natural language, and the system transforms the queries into structured SPARQL queries. The system attempts to formulate user queries but when there is encounters ambiguity**,** it uses clarification dialog to ask the user to manually disambiguate their query. The system also uses clarification feedback to formulate SPARQL queries, which is executed against the knowledgebase.

FREyA(Damljanovic, Agatonovic, & Cunningham, 2010) 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. The system uses semantically annotated ontology with syntactic parsing in order to formulate user natural language queries into structured SPARQL query language.

SWSNL is a work in (Habernal & Konopík, 2013) 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. The system processes queries using semantic analyses, and semantic interpretation to transform the user's natural language to SPARQL, and then retrieves result from knowledgebase. The system allow users to enter queries using keywords, complete sentences, or even paragraphs. The system was evaluated using two languages, mainly English and Czech. This system analyses user queries using natural language components, which are comprised of query pre-processing, name entity recognition and semantic analyses. The queries are formulated into SPARQL query language and a result is retrieved.

An Automated Semantic Query formulation approach was presented in the work of (R. Abdulkadir, R.A Yauri, 2017). There work automatically formulates user's natural language query to structured query based on using statistical machine learning approach. What differentiate their work with previous works is that there systems can translate paragraph length natural language query to structured query as against previous works that are mostly small fragment query. Additionally their disambiguation approach is capable of disambiguating words that are not found in WordNet. They used equivalent assertion to disambiguate non English words.

3.0 CONCLUSION

In Conclusion, semantic query formulation systems have been developed, ranging from manual semantic query formulation, through semi-automatic semantic query formulation, to automatic query formulation. The manual process requires a user to be familiar with the syntax of formal language or know what information is in the knowledgebase, which creates limitations in terms of how to make queries and what can be queried. Although the semi-automatic process provides some assistance to help the user to know what can be queried and how to make queries, it still requires a user to go through a lot by taking time during the query formulation process. Automatic semantic query formulation provides users with their favourite google-like search system, and in this case the system does not require heavy intervention by the user in the query formulation process. The automatic semantic query formulation approach saves users from having to be familiar with the complex syntax of formal query language or knowing the structure of the knowledgebase information, as in the case of manual semantic query formulation. Automatic query formulation also reduces the customisation and interaction processes a user must go through, as in semi-automatic query formulation. In automatic semantic query formulation a user's natural language query is transformed automatically into a structured query language such as SPARQL, which is then matched against the knowledgebase by several retrieval mechanisms in order to retrieve relevant information from the knowledgebase.

REFERENCE

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, *21*(3), 96–101. doi:10.1109/MIS.2006.62

G Kück, (2004), "Tim Berners-Lee's Semantic Web". South African Journal of Information Management.

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

Hu, Y. (2004). The Semantic Web : Current Status and Future Directions. [Power Point slides]. Retrieved from http://www.cs.nccu.edu.tw/~jong/pub/ mis0601talk.pdf.

Patel-Schneider, P. F. (2005): A Revised Architecture for the Semantic Web Reasoning. In: Proceedings of PPSWR'05, Dagstuhl, Germany.

Decker, S., Sintek, M., Billig, A., Henze, N., Harth, A., Leicher, A.Neumann, G. (2006). TRIPLE - an RDF Rule Language with Context and Use Cases. Proceedings of the W3C Workshop on Rule Languages for Interoperability. Washington DC. USA.

Bettina Fazzinga and Thomas Lukasiewicz. (2010). Semantic search on the Web. Semantic Web — Interoperability, Usability, Applicability. 89–96.

Tablan, V., Damljanovic, D., & Bontcheva, K. (2010). A Natural Language Query Interface to Structured Information, In ESWC 2010, volume 6088 of LNCS, pages 106{120. Springer, 2010 .

A.H.M. ter Hofstede, H.A. Proper and Th.P. van der Weide (1995) .Computer supported query formulation in an evolving context. In Proceeding Sixth Australasian Database Conference, ADC'95, Volume 17(2) of Australian Computer Science Communications, Adelaide, Australia (January 1995)

Blanco, RoiHalpin, HarryHerzig, Daniel M.Mika, PeterPound, JeffreyThompson, Henry S.Tran, Thanh. (2013). Web Semantics: Science, Services and Agents on the World Wide Web. Journal of Web Semantics.

Madhu, G., Govardhan, a, & Rajinikanth, T. K. V. (2011). Intelligent semantic web search engines: A brief survey. International Journal of Web & Semantic Technology, 2(1), 34–42. doi:10.5121/ijwest.2011.2103

Kharlamov, E., Giese, M., Soylu, A., Zheleznyakov, D., Bagosi, T., Console, M.,Waaler, A. (2013). Optique 1. 0 : 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.

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. and Goranov, M. (2003). KIM - Semantic Annotation Platform. In 2nd International Semantic Web Conference, Florida, USA, 2003 834-849.

Bellavia, G., Maio, D., & Rizzi, S. (1992). Resolving the query inference problem by optimizing query formulation cost, 1–30. Technical Report CIOC-C.N.R. n. 85.

Dongilli, P., Franconi, E. (2006). An Intelligent Query Interface with Natural Language Support. In: Proceedings of the 19th International FLAIRS Conference FLAIRS-2006, Melbourne Beach, Florida, May 2006.

Barzdins, G., Liepins, E., Veilande, M., & Zviedris, M. (2009). Ontology Enabled Graphical Database Query Tool for End-Users. In Eighth International Baltic Conference on Databases and Information Systems (DB&IS 2008), 105–116.

Guha, R., McCool, R., and Miller, E. (2003). Semantic Search. Proceedings of the WWW2003, Budapest, 2003.

GeneInfoViz. Gene Ontology. University Montpellier. Retrieved 6 October 2011 from http://irb.chu-montpellier.fr/fr/PDF/Bioinfo2007.

Gauch Groppe, J., Groppe, S., Ebers, S., & Linnemann, V. (2009). Efficient processing of SPARQL joins in memory by dynamically restricting triple patterns. Proceedings of the 2009 ACM Symposium on Applied Computing - SAC '09, 1231.

Tablan, V., Damljanovic, D., & Bontcheva, K. (2010). A Natural Language Query Interface to Structured Information, In ESWC 2010, volume 6088 of LNCS, pages 106{120. Springer, 2010 .

Stratica, N., Kosseim, L., & Desai, B. C. (2005). Using semantic templates for a natural language interface to the CINDI virtual library. Data & Knowledge Engineering, 55(1), 4–19. doi:10.1016/j.datak.2004.12.002

Zenz, G., Zhou, X., Minack, E., Siberski, W., & Nejdl, W. (2009). From keywords to semantic queries—Incremental query construction on the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web, 166–176.

Enikuomehin A.O., Okwefulueze D.O. (20120. An Algorithm for Solving Natural Language Query Execution Problems on Relational Databases. International Journal of Advanced Computer Science and Applications 169-175

Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., & Cimiano, P. (2012). Template-based question answering over RDF data. Proceedings of the 21st International Conference on World Wide Web - WWW '12, 639.

Wang, C., Xiong, M., Zhou, Q., & Yu, Y. (2007). PANTO : A Portable Natural Language Interface to Ontologies. Proceedings of the Fourth European Semantic Web Conference 473-487.

Aksac, A., Ozturk, O., & Dogdu, E. (2012). A novel semantic web browser for user centric information retrieval: PERSON. Expert Systems with Applications, 39(15), 12001–12013.

Mustafa Jarrar, Marios D. Dikaiakos. (2012). A Query Formulation Language for the Data Web," IEEE Transactions on Knowledge and Data Engineering 783-798.

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

Lei, Y., Uren, V., Motta, E. (2006).Semsearch: A search engine for the semantic web. In: Staab, S.,Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg.

Li, Y., Yang, H., Jagadish, H. (2005).NaLIX: An interactive natural language interface for querying xml. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 900–902. ACM Press, New York

Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., & Cimiano, P. (2012). Template-based question answering over RDF data. Proceedings of the 21st International Conference on World Wide Web - WWW '12, 639.

M. Moussa and R. F. Abdel-Kader. (2011).QASYO: A Question Answering System for YAGO Ontology. International Journal of Database Theory and Application Vol. 4, No. 2, June, 2011

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

Kaufmann, E., Bernstein, A., Zumstein, R. (2006). Querix: A natural language interface to query ontologies based on clarification dialogs. In: ISWC 2006.LNCS, 980–981. Springer, Heidelberg.

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

Ferrández, Ó., Izquierdo, R., Ferrández, S., & Vicedo, J. L. (2009). Addressing ontology-based question answering with collections of user queries. Information Processing & Management 45(2), 175–188.

Lehmann, J., & Lorenz, B. (2011). AutoSPARQL : Let Users Query Your Knowledge Base. In Proceedings of ESWC 1–15.

Deines, I., & Krechel, D. (2013). A German Natural Language Interface for Semantic Search, Semantic Technology. In proceeding of Second Joint International Conference, JIST 2012Nara, Japan, December 2012.

Yahya, M., Berberich, K., & Elbassuoni, S. (2011). 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

Damljanovic, D.; Agatonovic, M.; and Cunningham, H. 2012. FREyA: an Interactive Way of Querying Linked Data using Natural Language. The Semantic Web.125–138. Springer

Habernal, I., & Konopík, M. (2013). SWSNL : Semantic Web Search Using Natural Language. Expert Systems with Applications 40, 3649–3664.

Rabiah Abdul Kadir, Rufai Aliyu Yauri and Azreen Azman.(2018) Automated Semantic Document Retrieval. International Conference on Information Retrieval and Knowledge Management