CHAPTER 4

SEMANTIC SEARCH BASED QAAL SYSTEM USING QGT GRAPH MATCHING WITH SEMANTIC SIMILARITY

4.1 INTRODUCTION

The second phase of QA system is the query processing model in which the given query is matched with the terms used in ontology by using semantic pattern matching type. In question classification model, the given question is converted into query form and the answer is tested by using Q2Q algorithm. But, semantic pattern type of matching is not processed in it. In this chapter, pattern matching techniques are proposed and analyzed in given query in RDF graph based pattern matching technique based on similarity measurement analysis based on concepts, relations and graphs for matching the resultant query terms with the terms in knowledge base. The terms are framed as SPARQL queries in this thesis. In SPARQL query, four categories of query interfaces are used for retrieving answers from ontology. this issue and it is categorized as keyword-based, form-based, view-based and natural language-based systems. Among these, keyword-based system interface is mostly used. It is a challenging one to combine keyword-based interfaces with semantic
technologies. To achieve this task, semantic indexing is used with collection of representing indices.

In keyword-based search interface, two types of ambiguities are possible. One is lexical and another one is structural. Lexical ambiguities occur by homonyms and structural ambiguities occur when a sentence implies more than one meaning due to multiple assignments of same word. Lexical ambiguities can be solved by using information extraction and structural ambiguities are solved by adding phrasal expression.

For avoiding this type of ambiguities in keyword based search, semantic search is suggested and the features of semantic search are used to handle morphological variations and synonym with correct sense handling for retrieving correct answer. It also handles generalization for a QA system and concept matching handle, knowledge matching handle are also the features of semantic search. In addition it handles NL queries and questions and it has the ability to point out most relevant terms. Semantic search also has the ability to operate on user behavior and artificial mean.

In this thesis, QAAL graph traversal algorithm is proposed to match the patterns in query and the knowledge base. The performance of proposed QAAL system is compared with existing AQUALOG QA system based on precision, recall, F-Measure and MRR ratio. Syntactic based string metric algorithm is implemented in AQUALOG system. In proposed QAAL system, semantic string metric algorithm is suggested and semantic RDF graph

The organization of the chapter 4 is as follows: The survey of semantic search based QA system is discussed in Section 4.2. Pattern matching techniques and semantic search based matching terms are
explained in Section 4.3 and Section 4.4. Section 4.5 and Section 4.6 discuss about the reasons for graph representation and conceptual graph. RDF graph and semantic search matching with RDF graph are described in Section 4.7 and Section 4.8. QAAL graph traversal algorithm and usage of semantic search based search with QGT algorithm is QAAL system are described in Section 4.9 and Section 4.10. In Section 4.11, the performance evaluation and discussion is based on precision, recall, F-Measure and MRR values. Finally concluding remarks as summary is made in Section 4.12.

4.2 SEMANTIC SEARCH BASED QA SYSTEM

Vicedo & Ferrandez (2000) have discussed a QA system which is used with semantic representation of expected answer by using WordNet tool. This system is automatically generated a semantic context of the expected answer. A WordNet based tool is processed by terms for building its semantic context representation. This context is represented as weighted term vector. The semantic context of the answer is computed by adding the context vector of each term in question. This semantic context is used to identify the approximate expected answer type and subclassifying is allowed for each complex question. This semantic context based answer detection is used to improve the system performance.
for all semantic relationships in WordNet. Noun phrase chunk is also used in this model.

4.3 PATTERN MATCHING TECHNIQUES

Pattern matching is a technique for matching the query terms with knowledge base and retrieves the answer from ontology. Five types of Answer Extraction Pattern (AEP) matching techniques are used. These are simple pattern matching, surface pattern matching, POS pattern matching, syntactic pattern matching and semantic pattern matching types.

4.3.1 Simple Pattern Matching

Simple pattern matching scheme contains a question with simple terms like wh-form question and has a punctuation mark in it. This matching is used with the question with plain words, punctuation marks and question word. The exact question word and noun words are extracted from question and the remaining terms are extracted and implemented for matching patterns. The patterns have the form of sequences or tree structures. A tree pattern described a part of tree by starting with a node and specifying some branches and nodes.

4.3.2 Surface Pattern Matching
4.3.3 POS Pattern Matching

Part-Of-Speech is the process of marking up a word in a text based on definition and its context. It is used to find the pattern of relationships with adjacent and related words in a phrase, sentence or paragraph. Part-Of-Speech pattern matching scheme contains a regular expression in it. It is represented with symbols and meaning in it. If the regular expression contains ‘*’, it indicates none or any number of occurrences and ‘+’ indicates at least one or more number of occurrences. The noun phrase and verb phrase are used to set tag set for meaning.

4.3.4 Syntactic Pattern Matching

Syntactic patterns are represented with parse tree and parse dependency graphs. It is used as either parse trees or parse dependency graph for matching of terms with regular expression model. This type of parse tree is used to identify whether the phrase identified by named entity, numeric and temporal expression recognizer which occur at right position. The answer is expressed in a form of relative clause.

4.3.5 Semantic Pattern Matching

Semantic pattern contains POS tag, NE model, expressions, semantic entity and words in ontology. This pattern contains any number of slots with the above mentioned terms. These collections of patterns are filled with words from ontology.

Comparing the performance of all pattern matching types, semantic pattern is a challenging task and quite complex one. The ambiguity occurred in Q2Q model was solved in this proposed QAAL system by using semantic pattern type matching. Semantic pattern
matching can be implemented in three ways as graph matching, query expansion and RDFS reasoning scheme.

Graph matching mechanism is used in general form and it can be easily implemented in ontology. Graph matching in ontology is used for searching more complex object which is semantically closed to each other. Query expansion is a technique which aims to enhance the results of a search by adding terms for searching a query. Ontological query expansion is based on knowledge base in which the method to search for the terms that have to be added to the query. But, it is less developed in ontology field. RDFS reasoning scheme is based on graph matching with the nodes specified with RDF code. In QAAL system, RDF based graph matching technique is proposed and implemented. Because, the semantic pattern matching is implemented by using graph matching technique in ontology in a successful manner.

4.4 SEMANTIC SEARCH BASED MATCHING TERMS

Semantic search based analysis is introduced by Sheth et al (2004) to discover actionable knowledge from the ontology. It is essentially used to implement semantic search as a graph-theoretic based approach. It is used to discover and interpret complex relationships between resources. By modeling RDF database as directed graph, it is used to represent relationships between entities in knowledge base as graph paths with sequence of links.

A semantic matching approach in ontology based QA system is measured by the similarity between concepts. It is measured by the technique matchmaking which is divided into two types as syntactic matchmaking and semantic matchmaking. The structure of a task
specification is used for matching in syntactic matchmaking technique. In semantic matchmaking, the meaning and information content request is used for matching. For the semantic matchmaking scheme, ontology is used to find the similarities and to find the semantic distance between the terms. Most of the matchmaking schemes are mainly used in knowledge base for concepts of recognition of semantic matches. The matching process shall perform inferences and hierarchy calculation. For implementing this scheme, there is a need of providing Description Logic (DL) for representing the query terms in graph form. In this proposed model, RDF graph based representation is used.

4.5 REASONS FOR GRAPH REPRESENTATION

Graph representation is taken into account for the following reasons. Graphs are mathematical objects which are used for many tasks like visualization and analysis of data for humans. Graphs are a successful method to visualize and understand complex data. Robustness is possible in graph representation because it is possible to automatically produce a graph of any sentence. It is a time reduction work for constructing a graph representation for a given sentence. The size of a graph depends upon the concepts and relations available in a path. RDF is a language developed to annotate and describe information resources and their relations among each other. RDF allows the expression with highly interconnected collections of metadata assertions.

4.6 CONCEPTUAL GRAPH

Conceptual graph is introduced by (Sowa 1984) as a diagrammatic representation graph which contains logic forms. The vocabulary of a basic ontology is seen in graphical form. It contains
hierarchy of concepts and relations. Entities and their relationships are also included in this diagrammatic representation. This is encoded by labeled graph. This graph contains one node as concept node which is represented as entities and another node as relation nodes with relationships between entities or properties.

RDF graph is one of the types of conceptual graph. The concept nodes represent entities, attributes and events. The relation nodes identify the relationship between two concept nodes. With this representation, Part-Of-Speech tagger supplies each word with syntactic role tag. Syntactic parser generated its structured representation like noun phrase and noun. Semantic analyzer generates one or more conceptual graphs out of syntactic structure. With this concept type, nodes are represented with the elements by nouns, verbs, adjectives and adverbs. Then, the comparison of conceptual graphs should be checked for correction. It is in need to find the intersection of two graphs and to measure the similarity between two graphs with relative size for implementing this type of matching. This type of matching is processed in conceptual graph. But, there is a drawback in conceptual graph representation for matching query terms with terms in ontology. So, the representation of ontology with RDF graph is best suitable one.

4.6.1 Conceptual Graph Matching in QA System

Every conceptual graph has an entry and these entries are used for matching process. Similarity algorithm calculates similarity between sub graphs in recursive manner. Each relation associated with the entry induces a sub graph. It is suggested and proposed to find the best match from the combination of similarity algorithm. The matching ends when all the relations and concepts in query conceptual graph have been checked.
4.6.2 Benefits of RDF Graph

In RDF, a resource can be used as an instance of multiple classes. But, a unique concept type is used for each class in conceptual graph. The declaration of a resource as an instance of multiple classes in RDF is translated in conceptual graph model by generating the concept type with the highest common subtype of concept type. Conceptual graph has a unique relation type. In RDF, properties are binary type but in conceptual graph, properties are n-ary type. The n-ary type is used to represent relations among more than two objects. So, RDF graph is also represented as semantic graph model.

Semantic graph is a graph structured data representation which contains concepts as vertices and edges as relationship between concepts. Combination of vertices and edges in a semantic graph are types and attributes. Semantic graph is an associated ontology which specifies the possible concepts, relationships which is allowed in each pair of concepts and the attributes associated with each concept and relationship. Semantic matching approach attempts to match graphs based on their meaning as well as graph structure.

4.7 RDF GRAPH

The structure of any RDF is a collection of triples. This collection of triples are usually called RDF graph. Each triple states a relationship between two nodes in the graph. RDF has a semantic representation which provides a basis for reasoning about the meaning of an RDF graph. This is called as entailment. Entailment rules state the implicit information that can be inferred from explicit information. So, RDF query language contains an entailment. Data types are also supported
in RDF query language. RDF graph representation is easy because of the vertices in RDF. RDF graph is labeled from resource and most of edges have distinctive labels of triple.

4.7.1 RDF Statements

RDF statement contains three forms such as subject, predicate and object. It is normally called as triple. Subject is represented as concepts in ontology. Predicates are represented with property and the property values are specified as object in ontology. The property value can also be used as literal value. A literal is a string of certain data type and may occur as object of a statement.

RDF triple is a formation of subject, predicate and object. This RDF triples can be visualized as a directed labeled graph and it is mentioned in Figure 4.1. The figure shows subject and objects which are represented as nodes and predicate is represented as arcs. A set of RDF statements is called as RDF graph.

![Figure 4.1 RDF Statement Type](image)

4.7.2 RDF Schema

RDF Schema may be used to define classes and properties of ontology. RDF Schema is a specification which explains how to describe RDF vocabularies. It is a kind of system type for RDF.
### Table 4.1 RDFS Notations with Explanation

<table>
<thead>
<tr>
<th>S.No</th>
<th>RDFS</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RDF:type</td>
<td>Define a new class with type</td>
</tr>
<tr>
<td>2</td>
<td>RDF:Class</td>
<td>Define a new class</td>
</tr>
<tr>
<td>3</td>
<td>RDFS:Property</td>
<td>Define a property of new class</td>
</tr>
<tr>
<td>4</td>
<td>RDFS:Domain</td>
<td>Define a domain and values for property</td>
</tr>
<tr>
<td></td>
<td>RDFS:Range</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RDFS:Label</td>
<td>A string of text describing the resource</td>
</tr>
<tr>
<td>6</td>
<td>RDFS:Comment</td>
<td>Comment about a resource</td>
</tr>
<tr>
<td>7</td>
<td>RDFS:SeeAlso</td>
<td>Links to other relevant resources</td>
</tr>
<tr>
<td>8</td>
<td>RDFS:Literal</td>
<td>Primitive data type</td>
</tr>
<tr>
<td>9</td>
<td>RDFS:subClassOf</td>
<td>Define a new subclass for already existing class</td>
</tr>
<tr>
<td>10</td>
<td>RDFS:subPropertyOf</td>
<td>Define a new property for subclass</td>
</tr>
</tbody>
</table>

The list of RDF Schema syntax notations are mentioned in Table 4.1. Each notation is represented in RDF format as a schema form. This Table 4.1 is used to represent each notation with explanation in it.

RDF Schema does not provide actual application of classes and properties. It provides the framework to describe application-specific classes and properties. Classes in ontology can be related with each other by using rdf:subClassOf and rdf_subPropertyOf type. So, RDF description is synonyms to RDF graph and synonym to single RDF statement.
This proposed QAAL system, RDFS:subClassOf, RDFS:subPropertyOf, RDFS:Class, RDFS:Property and RDFS:Domain notations are used.

4.8 SEMANTIC SEARCH MATCHING WITH RDF GRAPH

There are two main ways of processing RDF graph patterns which are namely triple model and vertically partitioned model. In triple model, RDF pattern matching queries are processed as series of relations with triple relation as subject, predicate and object format. In this RDF, pattern predicate is used for representing property which is used for specifying relation between subject and object. In vertically partitioned model, results in a series of operations are processed with property based relations.

Semantic search matching is a technique which is used to find the similarity between the terms available in the given set of query with the terms stored in ontology. Before discussion about semantic search, RDF graph matching is necessary for checking the answer. In pattern recognition, graph matching problem is used to match a sample graph with the subgraph of large model graph where vertices and edges correspond to pattern and relation.

Two sets of similarity are available which are statistical similarity and semantic similarity. Statistical similarity uses statistical methods for a set of pattern of terms in query. Semantic similarity is used to compute the similarity of terms with meaning in it. The implementation
of RDF graph matching is suitable by calculating semantic similarity values among nodes.

4.8.1 Statistical Similarity

In statistical similarity based method Vector Space Model (VSM) is used to simplify the question to vector expression, characteristic value of vector and the pattern keywords of question. \( W_n \) is a set of words and all words included in questions are \( W_1, W_2, \ldots, W_n \). Each question expressed as n-dimension vector \( V \) which has a collection of Terms \( T_1, T_2 \ldots T_n \) and it is represented in Equation (4.1).

\[
V = < T_1, T_2, \ldots, T_n >
\] (4.1)

The computing method is represented as \( T_i \) which is expressed by Equation (4.2)

\[
T_i = p \times \log\left(\frac{T_Q}{q}\right)
\] (4.2)

Where \( p \) is the number of \( W_i \) words in the question, \( q \) is the number of questions obtained \( W_i \) in question and \( T_Q \) is the total number of questions.

4.8.2 Semantic Similarity

Semantic similarity is mainly computed by similarity of terms in query by meaning of terms. Three types of similarity are available which are surface similarity, structure similarity and thematic similarity. Surface similarity and structure similarity is based on matching of particular objects or relations. Thematic similarity depends on the
presence of particular patterns of concepts and relations. In this proposed work, it is considered a similarity between concepts, relations and conceptual graphs.

4.8.2.1 Similarity between concepts

In this proposed method, similarity between two concepts is obtained by the distance between them. The given two concepts are represented as $c_1$ and $c_2$. There is a need to calculate the distance $d_c(c_1, c_2)$. The similarity between two concepts is termed as $\text{sim}_c(c_1, c_2)$ which is defined with the inclusion of distance between two concepts and it is described by Equation (4.3).

$$
\text{sim}_c(c_1, c_2) = 1 - d_c(c_1, c_2)
$$

The distance between two concepts is calculated with the respective positions in concept hierarchy. In this proposed method, every node in concept hierarchy has a value which is obtained from the Equation (4.4).

$$
\text{nodevalue}(n) = \frac{1}{f^{d(n)}}
$$

Where $f$ is a predefined factor larger than 1 that indicates the rate at which the value decreases along hierarchy and $f$ will be set as 2 because the relation between two concept is analyzed at a time. $d(n)$ is the depth of the node $n$ in RDF graph where $n$ is the node in syntax tree. In this proposed system, it is decided to choose the longest path from the node to root for measuring it. For the root node, $d(\text{root})=0$. Here, root denotes the root node of a graph and $d(\text{root})$ is the depth of root node.
For the given two ontological concepts $c_1$ and $c_2$, the conditions are checked. The weight of a graph is represented as $w$. $c_0$ represents the closest concept for similarity calculation. The similarity measurement for two concepts of algorithm is described below.

if $c_1, c_2$ are the same concept, then $\text{sim}_c(c_1, c_2) = 0$
else if direct path relation between $c_1$ and $c_2$ is available, then $\text{sim}_c(c_1, c_2) = w[\text{subgraph}(c_1, c_2)]$
else if indirect path relation between $c_1$ and $c_2$ is available, then $\text{sim}_c(c_1, c_2) = \sum_{c \in \text{spath}(c_1, c_2)} w[\text{subgraph}(c_1, c_2)]$
else $\text{sim}_c(c_1, c_2) = \min\{\text{sim}(c_1, c_0)\} + \min\{\text{sim}(c_2, c_0)\}$

4.8.2.2 Similarity between relations

Similarity between two relations $r_1$ and $r_2$ is declared as $\text{sim}_r(r_1, r_2)$ which is described by the Equation (4.5).

$$\text{sim}_r(r_1, r_2) = 1 - d_r(r_1, r_2) \quad (4.5)$$

The distance between two relations ($d_r$) is calculated by their respective positions in the relation hierarchy. The relation hierarchy is constructed manually. So, the similarity between two relations in query ($r_Q$) and relations in resource ($r_R$) are defined by the Equation (4.6).

$$\text{sim}_r(r_Q, r_R) = 1 - d_r(r_Q, r_R) \quad (4.6)$$
The value is 1 when $r_Q$ subsumes $r_R$ and 0 otherwise. That is, the relation in query is a super type of relation in resource.

### 4.8.2.3 Similarity between graphs

Every resource has an entry of the graph and the users are required to set central word of query sentence that will be mapped to the entry of query. Each relation with the entry induces a subgraph. Similarity between conceptual graph consists of similarity between two entries and similarity between each subgraph pair. Similarity between any two subgraph $\text{sim}_G(c_Q, c_R)$ is also determined by the entries and their subgraphs according to their respective weights. $c_Q$ represents a concept in question and $c_R$ represents a concept in resource.

The symbol $r_Q^j, r_R^j$ denotes the $j^{th}$ relation with question and the $j^{th}$ relation with resource of the entry $c_Q, c_R$ where $j$ is the relation at particular position. To ensure the similarity between two graphs that will not exceed to 1. It is used to normalize the weights and the formula is described by the Equation (4.7).

For each subgraph with $c_Q$ as its entry with the relation $c_R$, the similarity between graphs is calculated. It depends upon the similarity measurement between two concepts $c_Q$ and $c_R$ and similarity measurement between two relations $r_Q^j$ and $r_R^j$.

\[
\begin{align*}
stm_G(c_Q, c_R) &= w(c_Q, c) \times \stm_G(c_Q, c_R) + \max\{\sum_j w(c_Q, j) \times \\
stm_r(r_Q^j, r_R^j) \times \left[ \stm_G(c_Q^j, c_R^j) \right] \times w(c_Q, c) + \sum_i w(c_Q, j) = 1
\end{align*}
\]

(4.7)
4.8.3 RDF Graph Pattern Matching

The standard query constructs for RDF data models is a graph pattern which is equivalent to the Select-Project-Join (SPJ) construct in SQL. A graph pattern is a set of triple patterns which are RDF triples with variables. There are two main ways of processing RDF graph patterns is based on usage of storage model. These RDF graph model is namely triple model and vertically partitioned storage model. In this model, triple relation is partitioned based on properties. In triple model, RDF pattern matching queries are processed as relational style on triple relation. In vertically partitioned approach results are in join operations with property-based relations.

4.9 QAAL GRAPH TRAVERSAL ALGORITHM

In the proposed QAAL system, ontology based RDF graph traversing is suggested by QAAL graph traversal algorithm. The algorithm is described in algorithm 4.1.

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Triple set model of query in tree form</td>
</tr>
<tr>
<td>Tg</td>
<td>Terms in triple set</td>
</tr>
<tr>
<td>QS</td>
<td>Set of SPARQL queries</td>
</tr>
<tr>
<td>SI</td>
<td>Simple term</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>Output in RDF graph form</td>
</tr>
<tr>
<td>TBV</td>
<td>To Be Visited node</td>
</tr>
<tr>
<td>VN</td>
<td>Visited node</td>
</tr>
</tbody>
</table>
Steps

1. Let TBV is needed to be visited in query
2. VN be the set of visited node list for given I which is in triple set model
3. If TBV is not empty, then
4. Assign SI as first element of TBV
5. Remove SI from VN
6. If SI is not in VN, then
7. For all terms in QS do step 8 to step 14
8. Term = Apply term to I
9. Add Term to TBV
10. O=O+Term
11. If Term is a concept, then
   Calculate sim_c(Term,VN)
12. If Term is a relation, then
   Calculate sim_r(Term,VN)
13. If Term is in a graph, then
   Calculate sim_G(Term,VN)
14. If SI is not in VN, then Goto Step 8
15. If TBV is not empty, then Goto Step 4
16. Else Stop the process
17. End If

Algorithm 4.1 QAAL Graph Traversing Algorithm
Function of QGT Algorithm in QAAL System

The input of this algorithm is a triple set model that contains user’s query which is converted from question model by using the proposed improved Q2Q algorithm. $T_g$ symbol is noted for this triple set model. The output of the algorithm is a RDF graph with exact terms in it.

Terms represented in input triple set $I$ as set as to be visited node for matching. Each term is input triple set $I$ is stored in Simple term $SI$ and it is matched with terms in ontology $ON$. If matching is available, the type of matching will be identified as concept, relation or graph type. Then, the corresponding semantic similarity measurement is calculated for concept, relation and graph. The term is added to output RDF graph $O$. This step is repeated until all the terms in query are visited. The output RDF graph $O$ is taken for answer extraction module.

A set of all queries are composed in knowledge base which have the same or correlative semantic type with user’s query. If the semantic type of users query and a selected answer type sentence are same, then same parts of semantic templates will be compared. Otherwise, their relevant parts of their semantic templates are compared based on a set of rules described in the section 3.3.3.

In this proposed work, the similarity between two concepts, relations and graphs are calculated. Matching of score between query and the answer of two are compared and the retrieved answer is more than one it is in ranked order. The matching is based upon RDF graph matching terms. The final result from the matching algorithm is the RDF graph based result which is finalized and listed in ranking order.
Every conceptual graph has an entry and the entries are used for matching process. It is in need to calculate similarity between two conceptual graphs. Each relation with the entry forms a subgraph and there is a need to find the best match from the given subgraph forms. Matching process is implemented when all the relations and concepts in query are checked.

4.10 USAGE OF SEMANTIC BASED SEARCH WITH QGT ALGORITHM IN QAAL SYSTEM

QAAL system contains three main modules such as inference engine, semantic search and answer extraction. An inference engine is a computer program which tries to derive answers from knowledge base. It is the brain of the system which is used to reason about the information in knowledge base for ultimate purpose of formulating new conclusion. It is considered as a special case of reasoning engine which is used for general methods of reasoning. This inference engine contains rules that are represented with predicate rules.

The semantic search model of QAAL system supports query template matching with similarity measurement technique. Semantic search model refers ontology for matching query generated from user question in ontology with the query model based on semantic template matching technique. The inference engine refers the knowledge base for checking the validity of retrieved answer with the help of semantic search model.

Answer extraction model is the important phase of QAAL system for ensuring the retrieved answer is in correct form or not. By checking the validity of answer, inference engine helps to match the
answer with semantic search model. Answer is ranked and verified by the user. This flow of concept is described in Figure 4.2

**Figure 4.2 QAAL System with Semantic Search Approach**

This figure shows the QAAL system with semantic search approach. The flow in the figure shows the importance of semantic search with the combination of inference engine. Matching of user question in query form is compared with the graph representation in ontology based on semantic similarity measurement. Semantic type of matching is done in this proposed QAAL system and the precision, recall ratios are analyzed and compared in QAAL system. Search types are compared with existing AQUALOG QA system. Evaluation is taken for QGT algorithm in QAAL system.
4.11 PERFORMANCE EVALUATION AND DISCUSSION

In this thesis, the evaluation of semantic search in QAAL system based upon precision ratio and recall ratio with RDF graph representation is implemented and analyzed with the help of proposed QGT algorithm with semantic similarity measurement based on concepts, relations and graphs. F-Measure value and MRR ratio both are also calculated as additional metrics for measuring the performance of the proposed system.

Evaluation is set for different sets of questions which are grouped as test sets. A test set is a collection of questions which are asked in different types of question form. Two test sets does not have the same pattern of questions in it. The semantic similarity measurement between concepts, relations and graphs is taken for evaluation of performance of proposed QAAL system and existing AQUALOG QA system. The evaluation is taken for proving the resultant RDF graph matching technique is a successful mechanism for proposed QAAL system.

4.11.1 Evaluation Based on Precision Ratio

To measure the performance based on precision ratio, 6 sets of test sets are taken with QGT algorithm in both proposed QAAL system and existing AQUALOG QA system.

The Figure 4.3 shows the comparison of the QAAL system with the AQUALOG QA system based on precision value with QGT algorithm. Test sets are grouped from simple question sets like what, define type and different type of complex question sets. The values are calculated based on satisfaction of user for the given answer to the question. The precision
values are calculated as a number of correctly answered questions by the number of returned answers.

![Graph showing comparison of precision ratio in QAAL and AQUALOG](image)

**Figure 4.3 Comparison of Precision Ratio in QAAL Vs AQUALOG**

The precision ratio is 0.3 in AQUALOG QA system for the test set five and the precision ratio is reached 0.8 in the proposed QAAL system for the same test set five. Test set five contains maximum number of complex type of questions in reason based and explanation based type in it. In other test sets, the precision ratio of proposed QAAL system is increased compared to AQUALOG QA system by approximately 0.4. The overall performance of QAAL system based on precision ratio is increased 2.42% compared to AQUALOG QA system. It is concluded that the QGT graph matching technique in QAAL system used to improve the performance of QAAL system to 2.42% as high precision ratio compared to AQUALOG QA system.
4.11.2 Evaluation Based on Recall Ratio

In the proposed QAAL system, the recall values are calculated for both proposed QAAL system and existing AQUALOG QA system based on semantic search with QGT algorithm. Recall values are taken as the ratio between the number of correct answers retrieved with the number of possible correct answers in the test set.

![Figure 4.4 Semantic Search with QGT Algorithm Recall Ratio in QAAL Vs AQUALOG](image)

The Figure 4.4 represents the comparison for semantic search with QGT algorithm in proposed QAAL system with the existing AQUALOG QA system based on the recall ratio. Different test sets have different recall ratios which is gradually high in each test. AQUALOG QA system follows keyword based search and semantic search with minimum recall ratio. So, the meaning of same word will be varied in different questions. But, results of these two questions return the same answer. In
this proposed semantic search based QAAL system, this drawback is reduced.

The recall ratio for the proposed QAAL system is 0.83 in test set five and the recall ratio is reached to 0.58 in AQUALOG QA system for the same test set. At the maximum, the recall ratio is reached to 0.94 in AQUALOG QA system in test set ten. By using QGT matching algorithm in proposed QAAL system, the recall ratio is reached to 0.95 for the test set ten. Comparison of all values getting from different test sets. It is concluded that the recall ratio value is improved 14% in QAAL system than AQUALOG QA system.

4.11.3 Evaluation Based on F-Measure Ratio

To evaluate the performance of F-Measure, precision and recall values are considered for QAAL and AQUALOG QA system. F-Measure value is based on precision and recall which is represented in Equation (1.6). Totally six test sets are taken for calculation to measure the performance of F-Measure metric. The calculated F-Measure values for proposed QAAL system and AQUALOG QA system are noted in graphical form in Figure 4.5.

The test sets for measuring the precision and recall metrics are taken for calculation of F-Measure ratio. In test set five, the F-Measure value is 0.40 in AQUALOG QA system and 0.81 values in proposed QAAL system which is comparatively high. In test set seven, the F-Measure value in AQUALOG QA system is 0.57 and QAAL system has 0.80. At the test set ten, AQUALOG QA system F-Measure is improved to 0.93 but QAAL system. QAAL system contains 0.97 F-Measure value for the same test set. Test set ten contains more number of factoid based questions in it.
By comparing all test set values, it is concluded that QAAL system performance is improved 19.3% based on F-Measure value than AQUALOG QA system.

4.11.4 Evaluation Based on MRR Value

MRR value is used to evaluate the system with set of answers returned and listed in ranking order. It is calculated as a multiplicative inverse of rank in first correct answer. In Figure 4.6, MRR ratio is noted for QAAL system and AQUALOG QA system. The figure shows the graphical values of QAAL and AQUALOG QA system MRR ratio with six different test sets from test set five to test set set ten.
Six different test sets are taken for measuring MRR ratio in QAAL and AQUALOG QA system. The MRR ratio at test set five in AQUALOG is 0.75 and QAAL is 0.76 respectively. Minimum change in ratio is found at test set five. For the test set six, MRR ratio is noted as 0.70 in AQUALOG QA system and 0.763 in QAAL system. The changes in MRR ratio is gradually increased in QAAL system compared with AQUALOG QA system. Among six test sets, test set ten gives a high MRR ratio in QAAL as 0.88 and for the same test set AQUALOG QA system has the MRR ratio as 0.87. It is concluded that compared to AQUALOG QA system, QAAL system performance is increased 2.7% (approximately 3%) based on MRR ratio by applying QGT algorithm in QAAL system.

Figure 4.6 MRR Ratio in QAAL and AQUALOG
4.12 SUMMARY

The goal of this proposed QAAL system based on QGT algorithm is used to improve the performance of QA system with better precision and recall values. This goal is achieved by implementing semantic search with semantic similarity measurement based on concepts, relations and graphs with improved query graph matching model based on similarity measurement between concepts, relations and graphs in this proposed QAAL system. For this type, RDF graph representation with annotation is set for each class and individual. Annotations take a major role used for representing the properties of each individual. So, the performance is improved to 10% in QAAL system.

Graph matching takes a major role for implementing semantic based search in question answering system. In this proposed approach, QGT algorithm is used for implementing graph matching based on RDF graph model with semantic search approach. QGT algorithm model is implemented and tested in proposed QAAL system and the existing AQUALOG QA system. Precision and recall ratios are considered as performance metrics for successful implementation.

The results are compared in both precision ratio and recall ratio. 3% is achieved in precision ratio of proposed QAAL system compared with AQUALOG QA system and 14% improvement is achieved in recall ratio of proposed QAAL system compared with the existing AQUALOG QA system. Additional metrics used for measuring the performance of QAAL system and AQUALOG system is taken as F-Measure ratio and MRR ratio. Based on F-Measure ratio, the QAAL system performance is
improved 19.3% compared with existing AQUALOG QA system. MRR ratio is based on ranking order of answers. So, MRR ratio is also improved approximately 3% in QAAL system compared with AQUALOG QA system.

It is concluded that QGT matching with semantic similarity measurement approach is used to improve the performance of QAAL system based on precision, recall, F-Measure and MRR ratios by using this type of evaluation.