

# **Cognitive Computing based Question-Answering System for Teaching Electrical Motor Concepts**

Atul Prakash Prajapati  
Electrical Engineering Department  
Dayalbagh Educational Institute  
Agra, 282005

D. K. Chaturvedi  
Electrical Engineering Department  
Dayalbagh Educational Institute  
Agra, 282005

## **ABSTRACT**

Today is the era of "Big Data", and one has to spend ample amount of time to extract the meaningful information from such a huge store of data. It leads towards such a question answering system which can offer exact and precise answers to user queries. For that there is a requirement of understanding user queries effectively. Thus this paper proposes a cognitive computing powered question answering system in the field of education, which possesses the power of Natural Language Processing (NLP). Here, cognitive computing provides the methods for synergism of several powers into a single architecture, NLP provides understanding of the user questions effectively, and Ontology endows with the techniques for the construction of robust knowledge base. So for the realistic implementation of the proposed architecture, the education domain has been chosen and will be teaching electrical motor concepts to the edification of the students.

## **General Terms**

Question-Answering System, NLP, Ontology.

## **Keywords**

Question-Answering (QA) System, NLP (Natural Language Processing), Ontology, Education Domain (Basic Electrical Motor Concepts).

## **1. INTRODUCTION**

Several approaches/tools have been proposed by the researchers over time for the effective utilization of the electronic data present over the web. Therefore several systems/tools/techniques like Intelligent Tutoring System (ITS), Expert Systems have been proposed. These systems were of specific nature which were able to do particular defined task; but these days there is a requirement of generalized systems. Therefore, there is a need of architectural way of designing the systems. Subsequently, the defined architecture should be of robust in nature which has the easy interfacing capability to other techniques. Here, the cognitive computing techniques provide such capability to the systems. Following section provides a brief introduction to the systems which were designed for the effective utilization and edification of the users. [1] proposed Semantic Network Based, NLP Powered Question Answering System for teaching Electrical Motor Concepts to the graduate students. [2] proposed "Electronix Tutor" for teaching

"STEM" concepts to the students. [3] proposed an involuntary appraisal system to appraise the quality of answers offered by the online health Q-A systems. They pointed out that the existing systems are lagging behind to provide quality answers to the community questions. So, they proposed a DBN (Deep Belief Network) learning framework for learning

the hidden semantics of answers in HQA (Online health question answering) systems. It further helps the classifier in the classification of answers. [4] proposed a knowledge-graph pedestal spoken question answering system. They established that the existing speech recognition systems barely confine the acoustic features and the language models for speech recognition. Throughout this process errors and inaccuracies are ignored which mortifies the system's performance. So they suggested that the combination of NLP and speech recognizer will surmount with these problems. As the NLP interface uses the knowledge graph based model and provides the semantic capability to the speech recognizer. [5] proposed a technique for finding allied queries from the question labels and its body for the community question answering knowledge bases. To accomplish the objectives he used NER (Named Entity Recognition) and NLP (Natural Language Processing) techniques and found that NER approach is the best in the optimization of the questions. [6] proposed a method for Text-Q-A systems, that exploit the dynamic memory networks for analyzing the questions. [7] proposed a hybrid querying method (SK-Query). This proposed method prevails over the problems present in existing querying systems, which is based on keyword and SPARQL querying. Although both the techniques are adequate but have some downsides, as keyword searching endures from the precision accuracy of searched results and SPARQL endures from the format issue. It sometimes faces intricacy in modeling proper query format for some queries. [8] proposed ASLEM (Adaptive semi supervised ELM) machine for the recognition of subjectivity of the questions asked in CQA's (Community Question-Answering). They resolute on two main problems, first one is data imbalance and the second one is labeling of data as it reasons a great problem while data is manually labeled. As an elucidation to this Gaussian modeling was proposed. [9] proposed an interactive dialogue based model that is for identifying customer behavior and indulgent to the needs of the online customers. This system is applied to an online Q-A system which considers yes-no types of questions and builds a model. This model additionally forecasts the conclusions for fulfilling customer needs. [10] proposed a cognitive system for involuntarily solving RAT (Remote Associates Test) queries. RAT problems are used to measure human's ingenuity by performing psychological tests. An empirical test is also performed in view of humans and the results are evaluated against the proposed system.

[11] proposed a trigger and class based language modeling technique for getting better sentence retrieval method throughout the process of question answering. To confine the relationship in between different terms (words) they used class model. This method uses the similar concept as used in the clustering algorithm, where similar words go for the same cluster. This clustering is further useful for searching germane

sentences. On the other hand, the trigger model forms a word couple (target and trigger words) throughout the training process. These pairs of words are included in the trigger model, which is further used in establishing the relation in between a question and a sentence. [12] proposed a Q-A system for answering the queries of biologists. They argued that the existing system uses search engines which consult an assortment of resources for discovering the answers. Thus it is a time-consuming approach as well as the conclusion is also not so precise. For this, they proposed a Q-A system that provides the accurate answers to the queries by consulting a variety of different type of resources. [13] proposed a key word based searching system that applies the power of Hidden Markov model for discovering the correct resources and the power of SPARQL querying for pronouncement of the correct answer from those resources. [14] proposed a system

that employs machine learning approaches (ANN and Clustering Algorithms) for discovering the correct resource persons to the online community Q-A systems. [15] proposed an Eclipse plug in based replay system which keeps track of the changes made throughout a software development process. This proposed system assists a programmer by providing the insights of changes made during a software development lifecycle. [16] proposed a language independent Q-A system. The proposed system overcomes the redundancy issue which exists while extracting data from the web.

This paper is organized in the subsequent sections: The introduction section of this paper discusses the concise survey of systems that have been designed in this area. Background section explains the overall architecture of the proposed system. Methodology describes the construction process of the proposed system by taking an example of question analysis and answer retrieval. Experimental design and results section shows the practical implementation of all the modules of the proposed system. The conclusion summarizes the work done in this paper. Rest sections discuss the challenges and future aspects that will be covered in near future.

## 2. BACKGROUND

Past systems were of specific natured system, which were able to do the particularly defined task. Also, the earlier web was of structured nature and the size of data was not too big. But, today everyone is dealing with other kinds of problems like today's era is of "Big-Data", where one has to deal with various types of data (Structured, Semi-Structured, and Unstructured) additionally the data volume is also very huge. One has to spend an ample amount of time just to extract meaningful information from such a huge store. Therefore there is a need of such a system which has the capability of understanding the situation and accordingly it should have the decision making capability. To surmount the problems present in the past QA systems this paper proposes a cognitive computing powered QA system for teaching the basic "Electrical Motor" concepts to the students, which will provide exact answers to user queries rather than providing the documents. Here, the cognitive computing provides easy interfacing facility for synergizing the several techniques into a single architecture which leads towards a generalized

system. The proposed QA system processes the user questions semantically rather than the keyword based searching. During this process the "NLP" function helps the QA agent. Here, NLP provides several functions like Tokenization, Part of Speech Tagging (POS), Named Entity Recognition (NER), Chunking, etc. For the implementation of a robust knowledge base, the agent follows the ontology approach (RDF/XML format for storing facts in knowledge-base) refer [17] for Ontology construction process.

### 2.1 Proposed Architecture

For explaining each module and construction process of this cognitive Q-A system, following assumptions have been made. This system has TSM (Teacher-Student Module), KMM (Knowledge Modeling Module), NQCM (NLP Question Corpus Module), QP (Query Processing Module) and RIM (Reasoning and Inferencing Module).

THEOREM 1: This anticipated system has following five tuples <TSM, QP, KMM, NQCM, RIM> refer

(Figure: 1).

- a) TSM - Teacher Student Module: This module subsists at UIM interface. In this module, a teacher or an expert stuffs the facts in knowledge corpus and in the question corpus also. Subsequently a student accesses the system using the "TSM" module and asks the queries related to "Electric Motor's" domain.
- b) QP - Query Processing Module: This module exists at UIM interface. This module handles the user queries. It implicitly uses reasoner and inferencing module for inferencing and establishing the relationships in between entities.
- c) KMM - Knowledge Modeling Module: This module is controlled by the TSM module. Using this module a teacher or an expert fills the facts in knowledge corpus.
- d) NQCM - NLP Question Corpus Module: This module is also controlled by the TSM module. Using this module a teacher or an expert fills the entities and associations in the question corpus.
- e) RIM - Reasoner and Inferencing Module: This module is implicitly controlled and used by the QP module.

THEOREM 2: The TSM module has two tuples <KMM, NQCM>.

$$\left\{ \begin{array}{l} KMM \in TSM, (Knowledge Modelling Module) \\ NQCM \in TSM, NLPQ-Corpus Module. \end{array} \right\} (1)$$

This TSM module is accountable for managing these two modules (KMM and NQCM). A teacher fills the facts in knowledge corpus and also sets up the question corpus (NQCM) for natural language processing. At the time of the query, if a value does not subsist in knowledge corpus, the teacher fills the required fact using the TSM module

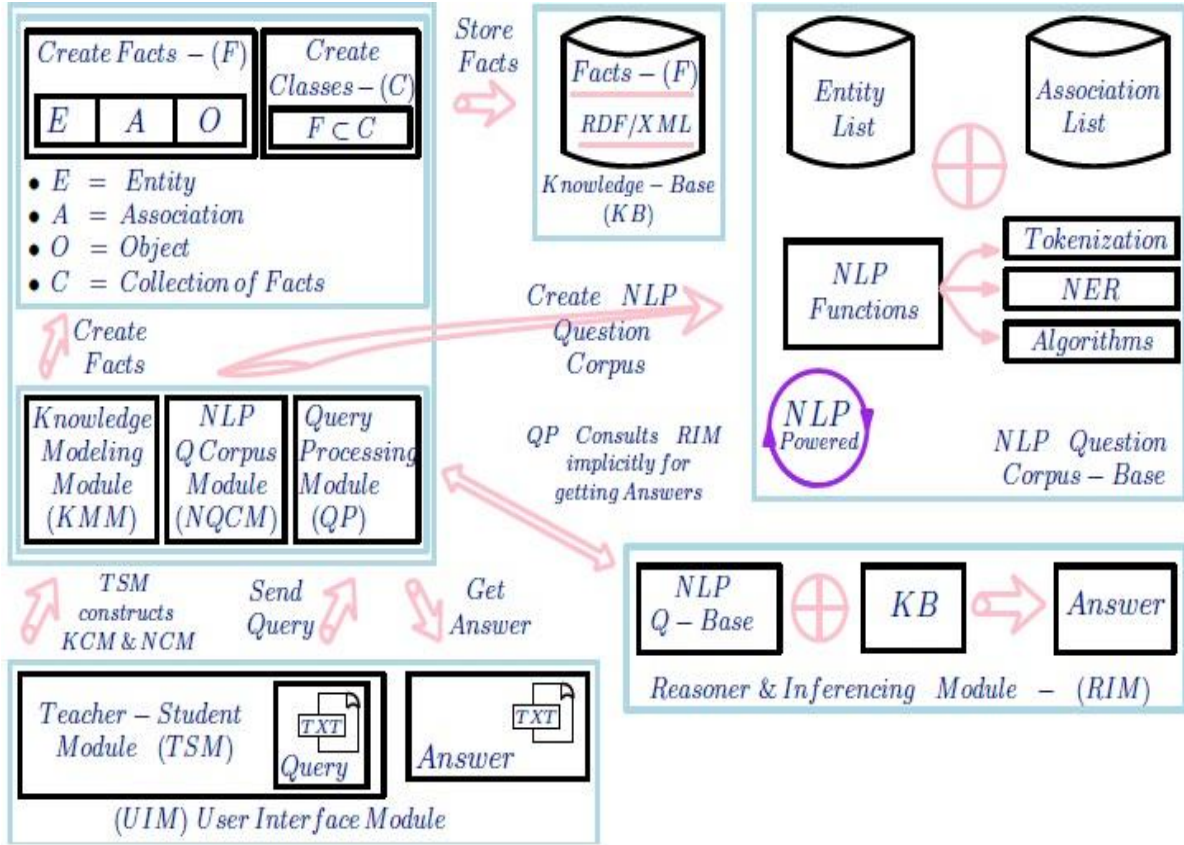


Fig 1: Schematic Diagram Showing the Architecture of the proposed NLP based Cognitive Question – Answering System.

THEOREM 3: The KMM module has following three tuples.

$\langle F, C, Ax \rangle$ .

$$F := \{(t) : t = \langle E, A, O \rangle : \begin{aligned} &(E \in \{Entity\ Set\}), \\ &(A \in \{Association\ Set\}), \\ &(O \in \{Object\ Set\}). \end{aligned} \quad (2)$$

A teacher stores the fact into the knowledge-base to represent knowledge. The fact is a triplet (entity, association, and object).

$$C := \left\{ \sum_{i=1}^n F_i : \forall (n > 0) \right\} \quad (3)$$

(C) is a collection of facts that belongs to same entity set but having different association and object.

$$A_x := \left\{ (A_i, rel(E_j, O_k)) : (i, j, k) > 0 \right\} \quad (4)$$

(Ax) is a set of axioms. These are the rules that tie the components of a triple together.

THEOREM 4: The NQCM module consists of following 4 tuples  $\langle SE, SA, SFN, SAlg \rangle$ .

$$SE := \left\{ \sum_{i=1}^n E_i : (\forall, i = [1, \dots, n]) \right\} \quad (5)$$

(SE) is a set of entities that are stored in NLP Q-Base entity list.

$$SA := \left\{ \sum_{j=1}^m A_j : (\forall, j = [1, \dots, m]) \right\} \quad (6)$$

(SA) is a set of associations that are stored in NLP Q-Base association list.

$$SFN := \left\{ \sum_{k=1}^o FN_k : (\forall, k = [1, \dots, o]) \right\} \quad (7)$$

This is a set of NLP functions, which consists of following functions (Split Sentence, Tokenization, NER (named entity recognition), POS (parts of speech tagging), chunker, etc.)

$$SAlg := \left\{ \sum_{l=1}^p Alg_l : (\forall, l = [1, \dots, p]) \right\} \quad (8)$$

This is a set of algorithms that are used, in the execution of NLP functions, and in mining the requisite values.

THEOREM 5: Query Module QP consists of following three

$$\langle Q, (NQCM, KMM) \in RIM, TSM \rangle.$$

tuples

$$\left. \begin{array}{l} Q = \text{Input Question.} \\ RIM = \text{It implicitly uses the} \\ \quad (NQCM, KB) \\ TSM = \text{If the value is not found in KB,} \\ \quad TSM \text{ module is informed for} \\ \quad \text{required action.} \end{array} \right\} (9)$$

THEOREM 6: Reasoner and Inferencing Module RIM consist of following two tuples  $\langle NQCM, KB \rangle$ .

$$\left\{ \begin{array}{l} NQCM = \text{NLP Question Corpus Module.} \\ KB = \text{Knowledge Base.} \end{array} \right\} (10)$$

## 2.2 Question Classifier

Proposed QA system can presently classify (W/H) type questions. Generally, the type of question helps in judging the answer type. As the interrogative pronouns have some inherent meaning. This system mainly focus on specific types of questions (related to the electric motor), like "What do you mean by induction motor". This implies that it is a definition type of question. So the Question classifier tries to get the intention of the user by analyzing the question. For this, there is need of NLP functions. At this point, Tokenization and NER (Named entity recognition) classifier play an important role. Following figure, (Figure: 1) demonstrates the question classification process. Here, the combination of interrogative pronouns and associations are used to explore the NLP Q-Corpus for getting the correct Q - Type.

Following algorithm (Algorithm: 1) describes the question classification process. Apply tokenization function on the input user query and collect the compulsory tokens. At this instant with these tokens, fill the Q-Template. If the template is filled and there is no missing token in the template, go for extracting the required QType from the NLP corpus. However, if there is any missing value; it declares that this is not a valid query.

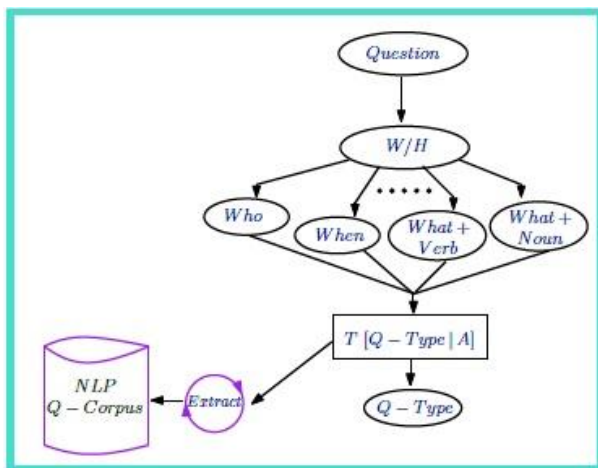


Fig 2: Schematic Diagram Showing the Question Classification Process.

Algorithm 1: Classification of questions.

- 1: procedure Query( $Q, Q\text{-type}$ )
  - >  $\left( \begin{array}{l} I/P : -(Q \in \text{Question/Query}) \\ \& O/P : -(Q\text{-Type} \in \text{Category}) \end{array} \right)$
- 2: Tokenization ( $Q$ )
- 3: fillTemplate (Format [ $IP|A$ ])
  - ( $IP \in \text{Interrogative Pronoun} \& A \in \text{Association Set}$ )
- 4: if (Template is formed Successfully) then
- 5: Search Required (" $Q\text{-Type}$ ")
  - > in NLPQ-Corpus.
- 6: return (" $Q\text{-Type}$ ")
- 7: else
- 8: " $Q\text{-Type}$ " = "Not a Valid QType".
- 9: return (" $Q\text{-Type}$ ").
- 10: end if
- 11: end procedure

Example 1: Following example (Example: 1) explains the Q-type selection process.

Steps:

1. Input NL Question.  
(What is the definition of induction motor?)
2. Tokenization and NER.
3. Frame the question.  
(What, definition)
4. Find Question-Type.  
(It is a definition-type question).

## 2.3 Component Extraction from the User's Question

After classification of the question, the next step is extraction of linguistic components from the input natural language (NL) query. Subsequently, try to extract linguistic components and fill the question template. This template takes the following form (Entity, Association, Object). It pursues the sentence structure (Subject, Predicate, Object). Rest tokens are discarded. The agent is concerned for the (Entity - Association) pair, as the object token is missing currently. The output of this phase has supplied as an input for the configuration of the query. In this step the agent is going to perform two basic operations.

- (1) Entity Extraction.
- (2) Relation Detection.

Following (Example: 2) and (Algorithm: 2) explains the extraction process of linguistic components (Entity, Association) from the user query.

### 2.3.1 Entity & Association Detection in the User's Question

**Example 2:** Following example explains the process of linguistic component identification from the user's natural language query.

**Steps:**

1. Input NL Question.  
(What is the definition of induction motor?)
2. Tokenization and NER.
3. Frame the question.  
(What, definition, induction motor)
4. Find out Question – Type.  
(It is a definition – type question.)
5. Fill the Q – Template (E | A | O).  
[Induction Motor | Definition | ?]

### 2.4 Query Generation

At this stage, the agent is ready to process the query. The output of the previous phase (linguistic searching), works as input to this phase. Therefore, the agent searches the knowledge-base for matching and extracting the necessary triples or fact. If the searched fact or triple is found extract the object (missing token of the template) and return to the user.

### Algorithm 2: Entity & Association Detection.

```

1: procedure Query(Q, (Format[E|A]))
   > ( I / P : - (E ∈ Entity Set) &
     > ( O / P : - (A ∈ Association Set) )
2: Tokenization (Q)
3: for (i = 0) to (Q – tokens.length) do
   if (token[i] ∈ E Set ( )) then
     fillFormat (Format [E | ?])
     > (E ∈ Entity Set)
     "Entity is Found".
     Set, (P == 1)
   > (P = for Checking Mis sin g Val)
   else
     "Entity is mis sin g".
   end if
end for
for (i=0) to (Q – tokens.length) do
  if (token[i] ∈ A Set ( )) then
    fillFormat (Format [ ? | A])
    > (A ∈ Association Set)
    "Association is Found".
    Set, (R == 1)
  > (R = for Checking Mis sin g Val)
  else
    "Association is mis sin g".
  end if
end for
if (P == 1) & (R == 1) then
  return (Format[E | A])
  "Both values exists".
else
  "Some values are mis sin g".
end if
end procedure

```

In case if the value is missing consult to your teacher (expert) for filling up the required information and return false "Value does not exist in the knowledge - base".

#### 2.4.1 Answer Searching & Extraction

**Example 3:** Following example (Example: 3) explains the answer extraction process from the Knowledge - base.

**Steps:**

1. Input NL Question:  
(What is the definition of induction motor?)
2. Tokenization and NER
3. Frame the question:  
(What, definition, induction motor)
4. Find out Question – Type.  
(It is a definition - type question)
5. Fill the Q - Template (E | A | O).

[Induction Motor, Definition, ?]

6. Search the Answer - Corpus.

[O = Definition (Answer)]

*Algorithm 3: Querying & Answer Extraction.*

```

1: procedure Query (Format [E|A|O], ?O)
  >  $\left( \begin{array}{l} I/P : -(E \in \text{Entity Set}), (A \in \text{Association Set}) \\ \& O/P : -(?O \in \text{Object Set}) \end{array} \right)$ 
2: Search (KB) > by using (E, A) pair.
3: if (Exists (Triple)) then
4:   Extract (?O).
5:   return (?O).
6:   "Object is found".
7: else
8:   "Object is missing".
9:   Consult (TSM)
     > (TSM) ∈ Teacher Student Module.
10: end if
11: end procedure

```

### 3. METHODOLOGY

#### 3.1 Ontology Representation

Here, the proposed agent follows the custom approach to triple construction. As, there is no set standard ontology available for the "Electric-Motors" domain. Therefore an ontology has created for the "Electrical Motor" domain using the "RDFSharp" library. The "RDFSharp" library integrates with the (C#) framework, which provides the friendly entry point for easy start playing with "RDF" and "Semantic - Web" concepts for ".Net" community users. One can integrate RDFSharp framework into ones project either by using "Nuget Package Manager" or by explicitly integrating it with the project from

(<http://www.nuget.org/packages?q=rdfsharp>).

For the validation of the constructed ontology, the agent uses

"RDFSharp ontology validator". It provides two types of validations "Implicit" and "Explicit" validations. In Implicit validation, taxonomies are automatically checked with the "OWL - DL reasoning" for preserving the consistency of the constructed ontology. In Explicit validation, ontologies can be validated through the "Validate" method. It provides the detailed report explaining the detected warning/error evidence. Following steps describes the detailed process of ontology development and querying process. Here, the agent supports the experience based knowledge base concept (where the domain ontology is created by the group of teachers) for answering the user queries.

#### Step 1: (Create Resources and Properties).

```

RDFResource Motor = new RDFResource
(RDFNamespaceRegister.DefaultNamespace "Motor");
RDFResource Induction_Motor = new RDFResource
(RDFNamespaceRegister.DefaultNamespace +
"Induction Motor");
RDFResource Wound_Rotor = new RDFResource
(RDFNamespaceRegister.DefaultNamespace +
"Wound Rotor");
RDFResource Squirrel_Cage_Rotor = new RDFResource
(RDFNamespaceRegister.DefaultNamespace +
"Squirrel Cage Rotor");
RDFResource definition = RDFVocabulary.FOAF.
DEFINITION;
RDFResource type = RDFVocabulary.RDF.TYPE;
RDFResource name = RDFVocabulary.FOAF.NAME;
RDFResource location =
RDFVocabulary.FOAF.LOCATION;

```

Firstly create entities and associations. Here a resource is an URI-named concept, modeled as an instance of RDFResource.

#### Step 2: (CREATE TRIPLES)

```

var Subject = new RDFResource (General_Ns_Sub + x) ;
var Predicate = new RDFResource (General_Ns_Pre + y) ;
var Object = new RDFResource (General_Ns_Obj + z) ;
RDFTriple triple = new RDFTriple
(Subject, Predicate, Object);

```



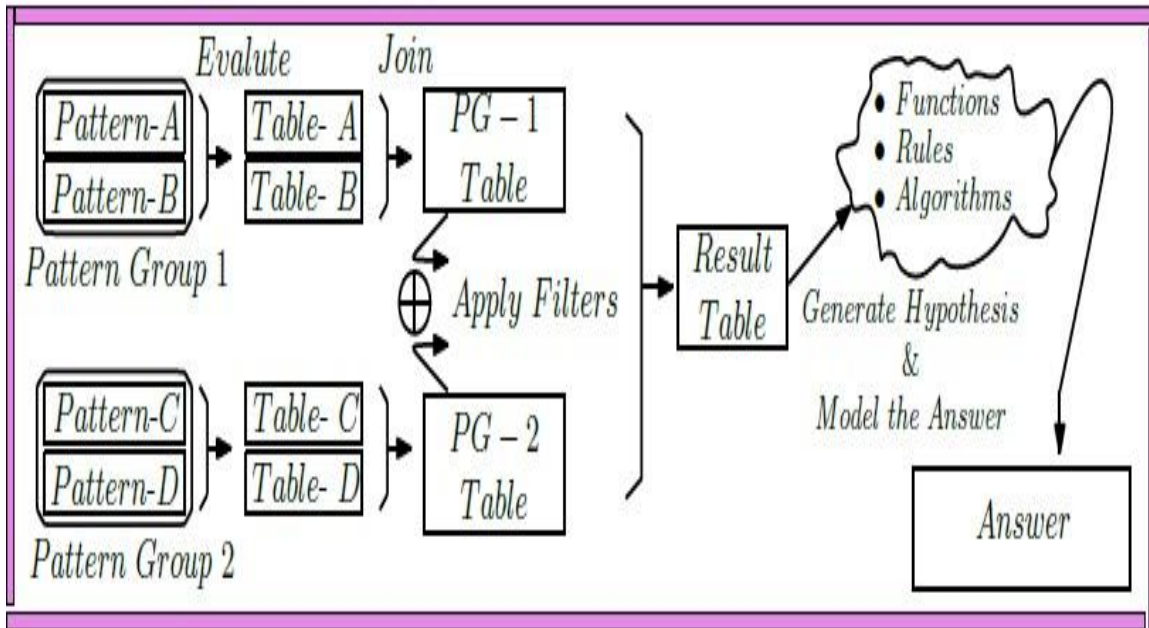


Fig 3: Algorithm for evaluation of SPARQL Queries.

A triple is an elementary assertion about a resource, modeled as an instance of RDFTriple. It is made up of a Subject part (the resource being described), a Predicate part (the resource being the verb of description) and an Object part (the resource, or the literal, being the knowledge described the subject).

**Step 3: (CREATE GRAPH)**

(A graph is an instance of RDFGraph; which is a URI - named Collection of triples modeled as instance of RDFGraph).

```
RDFGraph motor = newRDFGraph();
```

```
Motor.AddTriple (triple);
```

**Step 4: (EXPORT GRAPH TO FILE)**

(The content of a graph can be read from, or saved to, a RDF file or stream. RDFSharp supports following formats NTriples, Rdf/XML, TriX, and Turtle. This project uses Rdf/XML format).

```
motor.ToFile (RDFModelEnums.RDFFormats
```

```
NTriples, Environment.CurrentDirectory
```

```
+ \\motor1.nt);
```

```
motor.ToFile (RDFModelEnums.RDFFormats.
```

```
RdfXml, Environment.CurrentDirectory
```

```
+ \\motor1.rdf);
```

**Step 5: (CREATE MEMORY STORE)**

(RDFSharp provides an in - memory RDF store engine implementation which behaves like a graph, modeled as instance of RDFMemoryStore).

```
RDFMemoryStore memory = new RDFMemoryStore();
```

```
memory.MergeGraph (motor);
```

**Step 6: (PREPARE for QUERY)**

A SPARQL 1.0 engine (codenamed “Mirella”) is provided by RDFSharp. It allows creation and execution of queries over

Graphs, stores and federations:

CREATE SPARQL VARIABLES:

```
RDFVariable x = new RDFVariable (x) ;
```

- RDFVariable y = new RDFVariable (y) ;

**Step 7: CREATE SPARQL SELECT QUERY.**

```
SELECT DISTINCT ?X ?Y
```

Where {

```
#PG1
```

```
{
```

```
? Y <http://xmlns.com/foaf/association//hasRotor>
```

```
? X.
```

```
? X <http://www.w3.org/1999/02/22 -rdf- syntax-
```

```
ns#type >< http://MotorOntology.org/Sub/Rotor >
```

```
}
```

```
}
```

```
RDFSelectQuery q1 = newRDFSelectQuery ()
```

```
AddPatternGroup (new RDFPatternGroup (PG1))
```

```
AddPattern(new RDFPattern(y, hasRotor, x))
```

```
AddPattern(new RDFPattern(x, type, Rotor))
```

```
AddModifiers (new RDFDistinctModifier())
```

```
Add ProjectionVariable (x)
```

```
Add ProjectionVariable (y);
```

**Step 8: APPLY QUERY to the GRAPH**

```
RDFSelectQueryResults Result = q1.ApplyToGraph (motor);
```

**Step 9: Use Query results.**

Following algorithm, refer (Algorithm: 4), elucidates the querying process. It uses a function Query (Q, A), in which Q is a query requested by the user and A is the corresponding answer. First, analyze our question and fill the question

template. Subsequently, the agent verifies the absent values (if any attribute of the template is missing). It helps in choosing whether the asked question is correct or not. If the question is correct, the agent moves to the next level and fill the next question template (E, A, O). Further the agent explores the knowledge base using this next level template. If the match is successful, fetch the result and return back otherwise return false and confer with TSM module for filling the mandatory facts in the knowledge base. Here the experiential knowledge base has implemented within the proposed agent. A group of teachers has provided answers based on that the domain

ontology has been created. Therefore for each question there exists a set of answers.

*Algorithm 4: Querying Operation.*

```

1: procedure Query(Q, A)
> (I/P: -(Q ∈ Question) & O/P: -(A ∈ Answer))
2: Apply NLP_Functions(Q) > (Q ∈ Question)
3: Select (Tokens) & fill(Template(Format))
> (Tokens ∈ Q) & Format = [QType|E|A|AnswerType]
4: for (i = 0) to (i ≤ Template.Length) do
5:   if (Template[i] == Null) then
6:     Set, (Missing_Element ← 1)
7:   else
8:     Set, (Missing_Element ← 0)
9:   end if
10: end for
11: if (Missing_Element) then
12: Set A ← (Incorrect Question Plz Ask Again)
> (A ∈ Answer)
13: else
14: Select(Tokens) & Fill(Template(Q_Format))
> (Tokens ∈ Q) & Q_Format = [E|A|O]
15: Explore (KC) & Select (O)
> (KC ∈ Knowledge Corpus) & (O ∈ Object)
16: Set A = (O) > (O ∈ Object)
17: end if
18: Return (A) > (A ∈ Answer)
19: end Procedure

```

## 4. EXPERIMENTAL DESIGN & RESULT

This section shows the implementation of all the modules of the proposed agent. Here, the (Visual-Studio) IDE and (C#) language have been chosen for the execution of the proposed Q-A system and for the creation of knowledge-corpus, the RDF-Sharp framework has chosen. As RDF Sharp offers a friendly interface for modeling, storing, and querying with RDF data, and this framework is attuned with visual studio also. One can use NuGet for installing this framework ([http : //www.nuget.org/packages?q=rdfsharp](http://www.nuget.org/packages?q=rdfsharp)). For knowledge mining the (Linq to Rdf) querying language has been used. Subsequently, the NLP functions are used for the clear understanding of the user's questions.

### 4.1 Interface Module Designing

The proposed system mainly consists of following four components at the interface level.

1. TSM (teacher-student module) is accountable for addition and updation of facts (triples) in the knowledge corpus (KC). The proposed agent uses RDF/XML annotation method for storing the facts in KC, as it is encouraged by ontology way of storing knowledge. Contrasting rule based systems the proposed system is an autonomous system and for attaining the cognitive capability, it supports updation and amendment of facts as necessary. Also a student can ask the questions to the proposed agent using the TSM module.
2. NLP Q-Corpus Module (NQCM) is accountable for supervision of question corpus. Clear understanding of the question is the crucial part of any Q-A system. As the semantics of the question helps in finding the best decision and it also decides the performance and credibility of the system. NLP (Natural language processing) provides this capability to the proposed agent. Here, the NLP functions and algorithms use the question corpus for understanding the user query.
3. KMM (knowledge modeling module) is accountable for managing the knowledge base. Here, the Knowledge is stored in the form of triples (E, A, O). It is believed that a robust knowledge base is worth more than a good quality algorithm. Thus to achieve the cognitive computing capability one should use robust knowledge base. The proposed agent uses RDFSharp interface for modeling RDF triples and the set of triples forms up the knowledge-base. Subsequently they are represented in the form of domain ontology. Teacher works as an expert and stores the facts and he can also modify the existing triples and can add new triples as needed.
4. QP (query processing module) handles the user query. The agent uses (Linq to RDF) for getting the answers from the knowledge corpus.

### 4.2 NLP Q-Corpus Module designing

The proposed agent employees separate corpora for storing the entities, associations, and question type values. These tokens assist NLP functions and algorithms in understanding the questions. Following Figure (Figure: 4) shows the NLP Q-Corpus. It has separate corpus for representing Entity (E), Association (A), and (Q-Type) values.



### 4.3 Knowledge-Corpus Module Designing

The proposed agent uses RDFSharp interface for modeling RDF triples. A triple is a combination of three tokens (E, A, O) and the collection of triples forms the knowledge corpus. Following figure (Figure: 5) represents the knowledge corpus implemented within the proposed agent. The proposed agent uses the experience based knowledge base here the group of teachers are providing the answers for modeling the knowledge base of the proposed agent. Thus for each question there exists many answers in the knowledge base and the agent chooses the best answer for the particular question.

Fig 4: Schematic Diagram Showing the NLP Q – Corpus.

**Example 4:** The proposed agent uses following name spaces for representing the triples.

**Steps:**

1. Create Resources (Prefixes):
  - a. Motor Ontology Representation:

Motor: <http://www.motorontology.org/mtr#>

- b. Entity Representation:

E: <http://www.motorontology.org/sub#>

- c. Association Representation:

A: <http://www.motorontology.org/pre#>

- d. Object Representation:

O: <http://www.motorontology.org/obj#>

2. Create Triples:

< E, A, O >

3. Create Graph (G):

4. Add Triples in the Graph (G):

5. Add Graph into a Container:

6. Save the Container to a file:

### 4.4 Query Processing Module

The query processing module of the proposed agent uses LINQ to RDF, querying method for the extraction of knowledge. This module implicitly uses the RIM (Reasoner and Inferencing Module). The "RIM" module explores the KB and uses the concept of inheritance for establishing relationships in between entities.

**Example 5:** Following example explains the querying process. Subsequently, provide linguistic components (E, A) of the user query as an input to the query processing module QP.

**Steps:**

1. Input < E, A, O >
2. Create a Graph (g).
3. Load the Knowledge graph from file into g.
4. `Varm_s=g.selectTriplesbySubject (E)selectTriplesbyPredicate (A)FirstorDefault();`
5. `Var Object = SelectObject(m_s);`
6. `return (O = Object);`

### 4.5 Result

Following (Figure: 6) illustrates the practical implementation of the proposed agent's Q-A interface. A user query is being answered in this module. To achieve the abstraction all the functions are carried out in the background. Subsequently, NLP functions utilize NLP Q-corpus for understanding and extracting the question type. After getting the required question type the agent explores the knowledge base using query processing module QP. If the answer subsists in the knowledge corpus it is provided to the user; Otherwise, QP consults the TSM (teacher-student module) to supply the required knowledge. It is a cognitive system thus it allows to update and add new knowledge in its knowledge base. Here, the teachers help the agent by appending and updating new facts in the knowledge corpus.

**Example 6:** Following example explains the working of Figure: 6.

**Steps:**

1. Input User Query.  
(What is the definition of induction motor?)
2. Perform Tokenization.
3. Perform NER.
4. Extract the Tokens.
5. Identify the Q – Type.
6. Explore the Knowledge - Base for getting the answer.
7. Return answer to the user.

There is no benchmark questions set available for the evaluation of electrical motor's concept domain. Therefore, 15 sample questions have been taken for the assessment of the proposed QA system. The standard techniques like, Precision, and Reciprocal Rank concept have used to judge the retrieved answers. Here the system uses the concept of experiential knowledge base where the knowledge base is designed by the teachers and represented in the form of domain ontology. The group of teachers have provided the set of documents and the knowledge engineer has subsequently created the "Electrical

Motor's ontology". Each teacher has its perception, thinking, and way of expressiveness for a particular answer, thus there exists a set of solutions for each question. Based on this concept to judge the retrieved solutions, the P@5 (extracted five relevant answers in among of available responses) and RR (Reciprocal Rank Ratio) parameters have taken as a performance evaluation parameters for judging the performance of the proposed QA agent. The proposed QA system obtained good results (P@5 = 0.96, and MRR = 0.90). The proposed system finds out the precise answers to the questions rather than the finding the relevant documents which contains the answers. Following Table - 1 explains the P@5 and RR value for each sample question (15 questions/queries of electrical motor concepts).

**Precision (P):** Precision is the fraction of retrieved documents that are relevant to the query:

**Reciprocal Rank (RR):** The reciprocal rank of a query response is the multiplicative inverse of the position of the first correct answer.

$$RR = \frac{1}{rank_i} \quad (11)$$

**MRR (mean reciprocal rank):** calculates the average over a set of (n) queries.

$$MRR = \frac{1}{n} \sum_{i=1}^n \frac{1}{rank_i} \quad (12)$$

**Table 1: Table showing the evaluation results of the answers provided by the proposed QA system.**

Question No :	P @ 5	RR
1	1	1
2	1	1
3	1	1
4	1	1
5	0.6	1
6	1	1
7	1	0.5
8	1	1
9	1	1
10	1	0.5
11	1	0.5
12	1	1
13	0.8	1
14	1	1
15	1	1

## 5. CONCLUSION

Past Q-A systems were either rule based systems or their knowledge-base is designed in abstraction, that's why these systems can act upon better only under certain circumstances. Nowadays, everyone is talking about the intelligent machines, which can perform better in any condition. So to overcome the problems of past Q-A systems, this paper proposes a cognitive computing based (architectural way of designing the systems) QA system in the field of education to teach electrical motor concepts to the novice students. The fundamental purpose of proposing this architecture isto coalesce the power of cognitive computing with NLP to teach electrical motor concepts to the novice users.

1. The proposed QA agent follows System approach.
2. In this proposed architecture the agent currently can deal with W/H type questions.
3. NLP (Natural Language Processing) approach is proposed for the deep understanding of user's questions.
4. To overcome with the problems of rules based systems, the agent uses a dynamic approach to knowledge base construction; based on ontology.
5. The agent proposes the importance of teachers (by using the experiential knowledge base), who work as an expert also for checking the performance of the proposed system.

## 6. CHALLENGES AND FUTURE ASPECTS

This section of the paper discusses the problems and challenges that the present Q-A systems are facing and the future work to overcome these challenges. Nowadays both academic and industry personal are working progressively in the direction of self decision making systems. Still, they are not able to achieve the required target. This is because of the limitations of present technologies. Present technology can do the particular task effectively but as soon as there is a need of integration of different technologies for the construction of a generalized system, it fails or the performance compromises. Such generalized systems require the architectural way of designing the system which can accommodate different technologies within itself. Here the cognitive computing techniques play an important role. As, these techniques offers the synergism of different technologies within a single architecture like NLP, Computer vision, Machine learning, and many other technologies. Therefore, in the future proposed work, will try to provide additional cognitive computing features to the proposed agent.

## 7. DISCUSSION

1. **Experiential Knowledge based System:** The majority of the existing systems are using semantic web approach for data mining and answer extraction, so there is always be a concern of preciseness and authenticity of the provided answers. In the proposed agent the knowledge is provided by a teacher (domain expert). As humans do, in the case of any uncertainty, they generally rely on domain expert rather than searching information on the web.

2. **Robustness & Scalability of Knowledge-Base:** If one craft the knowledge-base of any system in abstraction, and it is not been exposed to the real world environment. Unquestionably, such systems will not perform better in every condition. The proposed agent is crafted deliberately under the

constraints of cognitive computing approach, thus its knowledge base module is adaptive. It faces the real world environment and in the case of any doubt, it requests the teacher and updates its knowledge.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:autoNS1="http://xmlns.com/foaf/association/" xml:base="https://rdsharp.codeplex.com/">
3 <rdf:Description rdf:about="http://www.motorontology.org/Sub/DC Motor">
4 <autoNS1:Definition rdf:resource="http://www.motorontology.org/01b/A DC motor is any of a class of rotary electrical machines that converts direct current ele
5 <autoNS1:Mean rdf:resource="http://www.motorontology.org/01b/A DC motor is any of a class of rotary electrical machines that converts direct current electrica
6 </rdf:Description>
7 <rdf:Description rdf:about="http://www.motorontology.org/Sub/Electric Motor">
8 <autoNS1:Definition rdf:resource="http://www.motorontology.org/01b/An electric motor is a device converting electrical energy into mechanical energy (usually
9 <autoNS1:Mean rdf:resource="http://www.motorontology.org/01b/An electric motor is a device converting electrical energy into mechanical energy (usually a torq
10 <autoNS1:Speed rdf:resource="http://www.motorontology.org/01b/The speed of a motor is the number of revolutions in a given time frame, typically revolutions p
11 <autoNS1:Types rdf:resource="http://www.motorontology.org/01b/Induction Motor, Synchronous Motor, Direct Current Motor." />
12 </rdf:Description>
13 <rdf:Description rdf:about="http://www.motorontology.org/Sub/Electrical Motor">
14 <autoNS1:Definition rdf:resource="http://www.motorontology.org/01b/An electric motor is a device converting electrical energy into mechanical energy (usually
15 <autoNS1:Mean rdf:resource="http://www.motorontology.org/01b/An electric motor is a device converting electrical energy into mechanical energy (usually a torq
16 <autoNS1:Speed rdf:resource="http://www.motorontology.org/01b/The speed of a motor is the number of revolutions in a given time frame, typically revolutions p
17 <autoNS1:Types rdf:resource="http://www.motorontology.org/01b/Induction Motor, Synchronous Motor, Direct Current Motor." />
18 </rdf:Description>
19 <rdf:Description rdf:about="http://www.motorontology.org/Sub/Induction Motor">
20 <autoNS1:Definition rdf:resource="http://www.motorontology.org/01b/An induction motor (IM) is a type of asynchronous AC motor, where power is supplied to the
21 <autoNS1:Mean rdf:resource="http://www.motorontology.org/01b/An induction motor (IM) is a type of asynchronous AC motor, where power is supplied to the rotati
22 <autoNS1:Rotor rdf:resource="http://www.motorontology.org/01b/Squirrel cage rotor and Wound rotor." />
23 </rdf:Description>
24 <rdf:Description rdf:about="http://www.motorontology.org/Sub/Motor">
25 <autoNS1:Speed rdf:resource="http://www.motorontology.org/01b/The speed of a motor is the number of revolutions in a given time frame, typically revolutions p
26 <autoNS1:Types rdf:resource="http://www.motorontology.org/01b/Induction Motor, Synchronous Motor, Direct Current Motor." />
27 </rdf:Description>
28 <rdf:Description rdf:about="http://www.motorontology.org/Sub/Slip">
29 <autoNS1:Definition rdf:resource="http://www.motorontology.org/01b/The rotor of an induction motor never rotates as fast as the stator magnetic field. This di
30 <autoNS1:Mean rdf:resource="http://www.motorontology.org/01b/The rotor of an induction motor never rotates as fast as the stator magnetic field. This differer
31 </rdf:Description>
32 <rdf:Description rdf:about="http://www.motorontology.org/Sub/Synchronous Motor">
33 <autoNS1:Mean rdf:resource="http://www.motorontology.org/01b/An alternating-current (A.C.) motor that runs at a speed that is equal to or is a multiple of the
34 <autoNS1:Definition rdf:resource="http://www.motorontology.org/01b/An alternating-current (A.C.) motor that runs at a speed that is equal to or is a multiple
35 </rdf:Description>
    
```

Fig 5: Schematic Diagram Showing the Knowledge-Corpus of the NLP based Cognitive Question-Answering System.

Fig 6: Schematic Diagram Showing the Q-A Interface of the Cognitive Question Answering System.

3. NLP Power: NLP provides the power of deep analysis and understanding of questions to the proposed agent. The clear understanding of the question helps in extracting the exact answer from the knowledge base.

## 7.1 Acknowledgments

I would like to thank Dr. M. B. Lal Sahab and Dr. P. S. Satsangi Sahab for his continuous inspirations and blessings.

## 8. REFERENCES

- [1] Prajapati A.P., Chandio A., Chaturvedi D.K. (2019), Semantic Network Based Cognitive, NLP Powered Question Answering System for Teaching Electrical Motor Concepts. In: Akoglu L., Ferrara E., Deivamani M., Baeza-Yates R., Yogesh P. (eds) *Advances in Data Science, ICIIT 2018, Communications in Computer and Information Science*, vol 941, 98-112, Springer, Singapore, doi: [https://doi.org/10.1007/978-981-13-3582-2\\_8](https://doi.org/10.1007/978-981-13-3582-2_8).
- [2] Graesser, A.C., Hu, X., Nye, B.D. et al., *ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics*, *IJ STEM Ed*, 5:15, 2018, doi: <https://doi.org/10.1186/s40594-018-0110-y>.
- [3] Hu, Ze and Zhang, Zhan and Yang, Haiqin and Chen, Qing and Zuo, Decheng (2017), A deep learning approach for predicting the quality of online health expert question-answering services, *Journal of Biomedical Informatics*, 71, 2017, doi:10.1016/j.specom.2017.05.001.
- [4] Jaya Kumar, Ashwini and Schmidt, Christoph and Köhler, Joachim (2017), A knowledge graph based speech interface for question answering systems, *Speech Communication*, 1–12, 92, 2017, doi:10.1016/j.specom.2017.05.001.
- [5] Figueroa, Alejandro (2017), automatically generating effective search queries directly from community question-answering questions for finding related questions, *Expert Systems with Applications*, 11–19, 77, 2017, doi:10.1016/j.eswa.2017.01.041.
- [6] Yue, Chunyi and Cao, Hanqiang and Xiong, Kun and Cui, Anqi and Qin, Haocheng and Li, Ming (2017), Enhanced question understanding with dynamic memory networks for textual question answering, *Expert Systems with Applications*, 39–45, 80, 2017, doi:10.1016/j.eswa.2017.03.006.
- [7] Peng, Peng and Zou, Lei and Qin, Zheng (2017), Answering top-K query combined keywords and structural queries on RDF graphs, *Information Systems*, 19–35, 67, 2017, doi: .1016/j.is.2017.03.002.
- [8] Fu, Hongping and Niu, Zhendong and Zhang, Chunxia and Yu, Hanchao and Ma, Jing and Chen, Jie and Chen, Yiqiang and Liu, Junfa (2016), ASELM: Adaptive semi-supervised ELM with application in question subjectivity identification, *Neurocomputing*, 599–609, 207, 2016, doi:10.1016/j.neucom.2016.05.041.
- [9] Stevens, Jon Scott and Benz, Anton and Reuße, Sebastian and Klabunde, Ralf (2016), Pragmatic question answering: A game-theoretic approach, *Data & Knowledge Engineering*, 52–69, 106, 2016, doi:10.1016/j.datak.2016.06.002.
- [10] Olteanu, Ana-Maria and Falomir, Zoe (2015), comRAT-C: A computational compound Remote Associates Test solver based on language data and its comparison to human performance, *Pattern Recognition Letters*, 81–90, 67, 2015, doi:10.1016/j.patrec.2015.05.015.
- [11] Momtazi, Saeedeh and Klakow, Dietrich (2017), Bridging the vocabulary gap between questions and answer sentences, *Information Processing & Management*, 595–615, 51, 2015, doi:10.1016/j.ipm.2015.04.005.
- [12] Neves, Mariana and Leser, Ulf (2015), Question answering for Biology, *Methods*, 36–46, 74, 2015, doi:10.1016/j.ymeth.2014.10.023.
- [13] Shekarpour, Saeedeh and Marx, Edgard and Ngonga Ngomo, Axel-Cyrille and Auer, Søren (2015), SINA: Semantic interpretation of user queries for question answering on interlinked data, *Web Semantics: Science, Services and Agents on the World Wide Web*, 39–51, 30, 2015, doi: 10.1016/j.websem.2014.06.002.
- [14] Procaci, Thiago Baesso and Siqueira, SeanWolfgang Matsui and Braz, Maria Helena Lima Baptista and Vasconcelos de Andrade, Leila Cristina (2015), How to find people who can help to answer a question? Analyses of metrics and machine learning in online communities, *Computers in Human Behavior*, 664–673, 51, 2015, doi:10.1016/j.chb.2014.12.026.
- [15] Hattori, Lile and D'Ambros, Marco and Lanza, Michele and Lungu, Mircea (2013), Answering software evolution questions: An empirical evaluation, *Information and Software Technology*, 755–775, 55, 2013, doi: 10.1016/j.infsof.2012.09.001.
- [16] Heie, Matthias H. and Whittaker, Edward W.D. and Furui, Sadaoki (2012), A Question answering using statistical language modelling, *Computer Speech & Language*, 193–209, 26, 2012, doi:10.1016/j.csl.2011.11.001.
- [17] Prajapati A.P., Chaturvedi D.K. (2017) *Ontology Based Knowledge Representation for Cognitive Decision Making in Teaching Electrical Motor Concepts*, Silhavy R., Senkerik R., Kominkova Oplatkova Z., Prokopova Z., Silhavy P. (eds) *Artificial Intelligence Trends in Intelligent Systems. CSOC 2017. Advances in Intelligent Systems and Computing*, 573, 43-53, 2017, Springer Cham, doi: 10.1007/978-3-319-57261-1\_5.