A Comparative Analysis of Permutation Combination Based and Grammatical Rule Based Knowledge Provider System

PRASENJIT MUKHERJEE AND BAISAKHI CHAKRABORTY
Department of Information Technology
National Institute of Technology
Durgapur, India
prasen.mscit09@gmail.com
baisakhichak@yahoo.co.in

Abstract: - Permutation Combination (PC) based parsing algorithm of Knowledge Provider System (KPS) and Grammatical Rules (GR) based parsing algorithm of Automated Knowledge Provider System (AKPS) has been compared using the Time complexity, Space complexity and some other criteria. “Big-O” notation has been utilized by the Time function T (n) and Space function DSAPACE (T, X) to describe the time complexity and space complexity of both algorithms. Further it is clarified that PC based parsing system (KPS) is a subset of GR based parsing system (AKPS). The function T (n) and T (n, m) represents the equation of both algorithms of Knowledge Provider System and both this equation has been used in mathematical optimization to prove that one is subset of another.

Key-Words: - PC based KPS, GR based AKPS, Time Complexity, Space Complexity, Knowledge Provide

1 Introduction

Knowledge Management System (KMS) works on the principle of tacit knowledge to explicit knowledge conversion. There are many types of KMS and different types of algorithm have been used in conversion of tacit to explicit knowledge. A KMS may not be efficient due to the poor performance of an algorithm. So, efficiency of an algorithm has taken an important role in system development. Efficiency of an algorithm depends on many factors like high-speed processor, RAM capacity, Operating System used, quality of code generator by the compiler, etc. These factors produce the actual absolute time taken by the algorithm. If factors are changed, then definitely absolute time will be changed. Thus efficiency measurement of an algorithm by the absolute time is not useful in this respect. The Time complexity and Space complexity are applied for efficiency measurement of an algorithm. Knowledge is the internalization of information, data, and experience. Knowledge Management (KM) is seen as a great tool for improving and optimizing the functioning of for performance enhancement and several organizational knowledge models have been proposed and discussed from software modeling, domain ontology and semantic ontology perspectives as in [2] to improve organizational performance. We have introduced two KMS algorithms for tacit to explicit conversion in our earlier papers [3] [4]. We have introduced a KMS system in [3] controlled by individuals termed Knowledge Workers (KWs) that handles natural language queries from the users and provides them responses. The natural language queries may be in interrogative or assertive. The natural language query is tokenized and then, parsing technique is applied which is Permutation-Combination (PC) based. We have represented an automated KMS System in [4] where the algorithm first checks the user query whether is it interrogative or assertive. If the query is in interrogative then in parsing method automatically interrogative sentence grammar rules will be created and the interrogative query will be passed through these generated grammar rules. After this the semantic table will be produced which generate the conceptual form then logical form and finally physical form of database. An assertive sentence shall be handled in a similar way. A web-log based KM model has been discussed in [5] for knowledge creation and sharing. Architectural models of KMS have been discussed in [6] using Service-Oriented Architecture methodology on open and distributed environment. The proposed second KMS model is fully automated and individuals' means Knowledge Workers (KWs) concept is removed. This system also follows the request-response model. The same procedure will be adopted for the assertive query. Machine Learning is used for the entity extraction in academic research, rule-based systems dominate in commercial use and these rules are very effective, interpretable, and are easy to customize as in [7]. Intelligence is required for any NLP based system and it should be able to ascertain the meaning of a word or semantic. If the system is used as syntactic system, the semantic linked with the words needs to be represented as syntax in [8]. This paper describe the two KMS system models (KPS and AKPS respectively) and performance comparison of these two KMS systems algorithm based on criteria like Time Complexity and Space Complexity. For the time complexity the T(n) function has been used where n is number of tokens produced from natural language query. The following sections of this paper has been described below- Related Works in Section 2, Comparative Analysis in Section 3, Architecture and Algorithm of KPS in Section 3.1, Time & Space complexity in Section 3.2, Architecture and Algorithm of AKPS in Section 3.3, Time and Space complexity of AKPS in 3.4, Time complexity Graph of KPS and AKPS in Section 3.5, Space Complexity Graph of KPS and AKPS in Section 3.6, KPS system is a subset of AKPS system in Section 4, Maximum and Minimum value of a function f(n) of KPS in Section 4.1, Maximum and Minimum value of a function g(n) of AKPS in Section 4.2, Graph representation of Maximum and Minimum values of f(n) and g(n) in Section 4.3, Conclusion and Future work in Section 5.

2 Related works

Wibisono developed Natural Language Interface to Database is a users tools to access data from database using natural language in 2013. The development was based on domain independent NLDBI where the tool was Stanford Dependency Parser. A question type query handling and unit conversion has done as further enhancement in NLDBI. In new approach the NLDBI able to process the question type query and a library called JScience has been used for the unit conversion. Finally the proposed system creates the query and retrieves data from the database to the user as in [9]. A complete literature survey shows that lack of Information Communication Technology (ICT) knowledge and cost is a challenge in small and medium enterprises. It is stated in this survey that most important significance is system requirement specification in natural language (SRS-NL) to design a database. Basically the SRS-NL is the basic concept of Database conceptualization. The tool is very much needed for the non technical people may able to produce a conceptual model of database automatically using SRS-NL as in [10]. A syntactic or a shallow semantic analysis in natural language processing is called dependency parsing. The author studied an interface between dependency and stress. The statistical analysis has been done on stress distribution pattern across 24 dependency relations based on HIT (Harbin Institute of Technology) dependency scheme. The study shows that there is an inherent association between stress distribution and dependency relation as in [11]. An integrated Visual Analytics tool is java based which is used to extracts important keywords, relation and events from various different type text data sources using ontology and natural language processing method. This tool provides a search interface for efficient investigation to the user in large and complex data set as in [12]. Natural Language Interfaces (NLIs) are used to access semantically structured data. NLIs support restricted language to handle natural language ambiguities. The challenge was how a user can easily use the language to convey themselves within constrains imposed by the system. Two methods have been investigated for the improvement of the habitability of a natural language interface. The methods are stated that feedback and clarification dialog. Feedback method has been used to interpret the query where as clarification dialog used to control the query interpretations as in [13]. Interaction between computers and natural languages is a main goal of natural language processing. In present day most of the data is in unstructured form and very hard for the computers to understand for further process. Analysis is needed on this unstructured data to be converted into the structured form after defining the sentence boundaries, word boundaries and context dependent character boundaries. This new system stated a component based domain independent text analysis system which process the natural language and used to convert the unstructured form of data to a structured form as in [14]. Knowledge management system is a tool which follows the Request- Response model. Extraction of tacit knowledge to explicit knowledge in KMS model where request-response method is Tag-based and this method has been proposed and discussed in [15]. An application on rules based system stated the features of the language, applicability of rules and applied the neural network model for ambiguities. The Implementation of this application is pAninI based semantic analysis where the system follows the hybrid model and it is incorporated with the features of rules and neural network based as in [16]. A natural language interface may be attached with the Database, KMS or any other. K. Englmeier, J. Pereira, and J. Mothe has been introduced a system called WS- Talk system in his paper. This system has an ability to compute the relationship of mapping between the business logic and natural language story book which is described by a process designer as in [17]. Using natural language, Lilac A. E. Al-Safadi explains that a semi- automated system which creates the database from very early stage of database development. System uses the natural language text as a source to produce the conceptual model of database in [18]. There are many approaches on parser development. The top-down parser, bottom-up parser, deterministic parser are few most popular parser which works inside the NL Interfaces. In this approach Syntactic analyzer has been considered as a top-down parser and uses the context free grammar (CFG). Another model has been introduced on prototype implementation on the CLUO- an Open Source Intelligence (OSINT) system. The main work of this system to extracts and analyzes a huge amount of openly available information which is published on the internet as in [19]. Named Entity Recognizer for Filipino Text Using Conditional Random Field (NERF-CRF) is actually the study to focus on the creating a system which can identifies and classifies named entities present in corpus. Many different type modules have been used including a tokenizer and a part-of-speech tagger. The main approach in the system to use the conditional random field for the classification of identified named entities. The system is tasted by the Filipino biographies as in [20]. A new system has been designed to promote the feasibilities of the Knowledge Management and presenting knowledge by using computer aspect and shows an important of learning mechanism. The model is constructed by the information technology, the model can extracts data and infer some formulas in [21]. Many methods have been described linguistic features such as part-of-speech, syntactic structure and semantic qualities and some of the methods with common statistical measures such as term frequency and inverse document frequency as in [22]. The Arabic question answering system is developed on lightweight semantic–based system. The Arabic question answering system is based on paragraph retrieval. The framework is an explicit unification and this framework based on semantic similarities and query expansion (synonyms and antonyms) as in [23]. Architecture of factoid question answering system follows a framework and implementation of this framework based on the DBpedia ontology and the DBpedia extraction. This system used SVM learning machine algorithm to gain a high accuracy ratio as in [24]. The mapping of natural language construct to relational algebra using the E-R diagram may be a good methodology to represent the logical form of database as in [25]. For most of the KMS or Database system, a human-computer communication is important feature. The visual query system (VQS) is a tool which is a visual representation of domain of interest and expresses related queries and VQSs provide both a language to express the queries in a visual format and a variety of functionalities to facilitate user-system interaction as in [26]. Maximum NLIDB uses the Syntax-based system. The syntax-based system described in [27] that natural languages queries from client side are parsed.. In this approach the semantic interpreter also follow some rules to transform the parse tree to the intermediate logic query. A number of architecture has been developed on Natural Language Question Answering (NLQA) system. NLQA architecture is defined by the four basic modules which are suitable for enhancing current QA capabilities with the ability of processing complex questions. The first module classifies and analyzes the question. The process of retrieving the applicable documents is allowed in the second module. The processes started to retrieve documents in third module. Finally the extraction and response generation is performed in the fourth module. The NLP technique is used for processing the question and documents and extracts the answer as in [28]. A new system stated a concept where it finds the sense of a word in an electronic data. Users may use some words called slang in nature in electronic communication. This system uses the supervised learning procedure to detect that type of abusive words. The system also can detect some abbreviated forms using the semi supervised learning procedure. The system completely evaluates the slang words and the probability of a suspicious word to be a slang word as in [29]. The UT Austin system is built on the output of an existing relation extractor by augmenting relations. The relations have been described explicitly in the text which is indirectly from the stated relations using probabilistic rules. These rules are well-read from linked open data and these are encoded in the form of Bayesian Logic Programs [30]. Another good approach has been described in Stanford's participation in the French-English and English-German tracks of 2014 Workshop on Statistical Machine Translation (WMT). WMT has been used for large number of sets of features, word classes and an optional unconstrained language model. The improvement has been done on basic Phrasal system with sparse features over class-based language models, word classes and a web-scale language model as in [31]. This paper makes a comparative analysis of two KMS tools, the PC based KPS as in [3] and the GR based AKPS as in [4]. The comparative analysis has been done on the basis of time complexity and space complexity with a vision to develop a hybrid knowledge provider system (HKPS) to handle a wide range and variety of queries that may comprise of few token to large number of tokens.

3 Comparative Analysis of PC based KPS and GR based AKPS

The PC based KPS with Knowledge Workers (KWs) as discussed in [3] and GR based AKPS as proposed in [4], both process natural language queries and retrieves knowledge data from knowledge database. These Systems create the conceptual form of database to retrieve knowledge data from knowledge database. The comparative analysis of both systems has been done on the basis of Time Complexity, Space Complexity and on Advantages & Disadvantages of both systems. Finally, it has been proved that PC based KPS is subset of GR based AKPS. Hence, we would to able to develop an automated hybrid knowledge provider system in future, where, as per requirement the system may be able to automatically select its mode of operation. For a smaller system handling few tokens, it can work on KPS mode and for larger system handling many tokens, it can switch over to the AKPS mode.

3.1 PC based Knowledge Provider System (KPS)

The architecture of Knowledge Provider System follows the request-response model where each request will be in natural language. KPS receives each request termed query from the client side and checks its validation. The KPS has default database containing certain predefined queries and their responses. If the response to a client query is not generated from the default database in KPS, then the response is generated directly from the Knowledge Workers (KWs) as in **[3].** Direct response from KWs may be slow, but it helps in populating the DB of the KPS as the new Responses to unknown queries by the KWs are stored in the DB of the KPS as in [3]. The response from DB is faster than the response from Knowledge workers. KPS reads each query in natural language and analyzes NL query syntactically and creates the conceptual model of database (Entity- Relationship Diagram) from which the logical and finally physical model of database is generated.

NL Query →Syntactic Analysis →Processing →Conceptual Model →Logical Model → Physical Model.

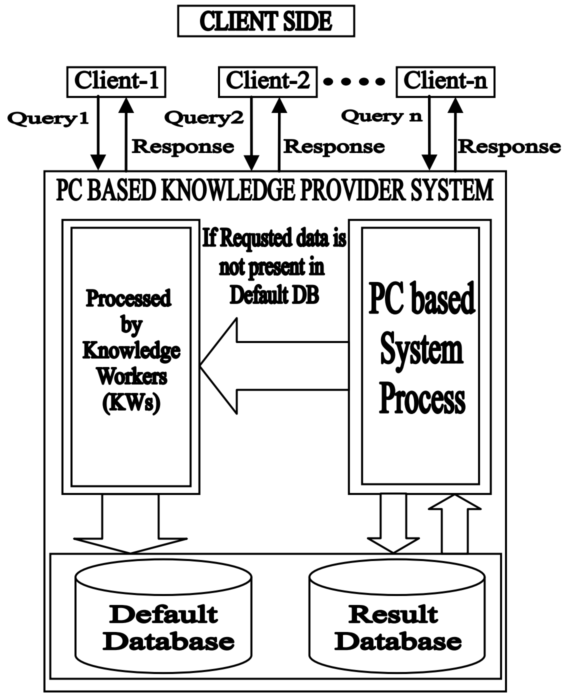

Fig.1: Architecture of KPS

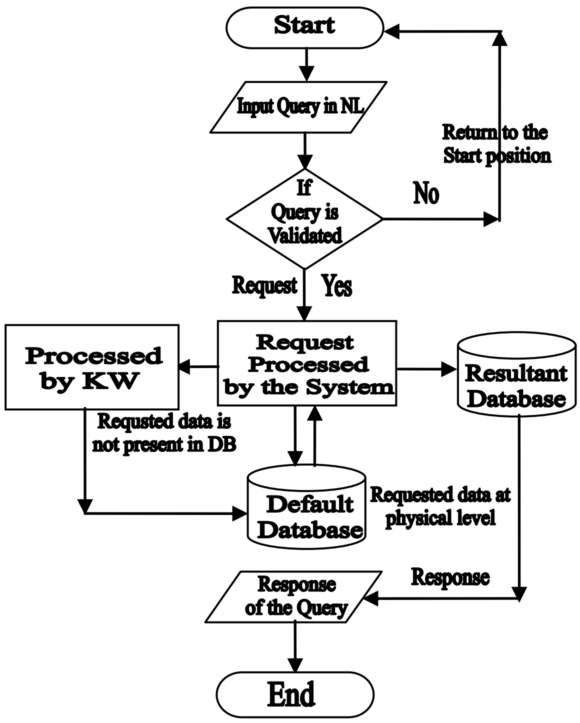

Fig. 2: Flow Chart of KPS

3.1.1 Algorithm

i. User posts Query in Natural Language.
ii. Query is checked for validity. If valid then go for further Processing, if not, then prompt again.
iii. The Query in Natural Language is tokenized. Query (String) = {$S_1, S_2, S_3, S_4, \ldots\ldots S_n$.} Where, S_1, S_2 are tokens.
iv.Each token is checked with English Grammar, placed in a Grammar Table (except for unique Tokens)and removed from the Token List after its placement in the grammar Table.

```
For i=0……n
{
If( Si== wh[])
 // Insert s1 into the grammar table in its position.
    Else
   If(Si == Pre[])
 // Insert S3 into the grammar table in its position.
    Else
      If(Si == Aux[])
      // Insert S3 into the grammar table in its position.
      Else
      If(Si == Det[])
      // Insert S3 into the grammar table in its position.
      .
      .
      If(Sn== Con[])
      // Insert S3 into the grammar table in its position.
      }
```

v. Check again for all unique Tokens (Words) and if the Unique Token matches with the content of Grammar table, then it is removed from the Token List. The Algorithm is applied to the rest of the Token List in a similar manner.
If unique Token does not match with the grammar table, then apply the Algorithm on Token List.
vi. After applying the Algorithm will get the Conceptual form of Database (E-R representation) from the Natural Language Query.
vii. If Database and Tables already created response the Query else create the Database and Tables and send query to the Knowledge Workers (KWs). Knowledge Workers will update the Database from where the response will be generated.
viii. Resultant Database stores the response from where the user will get the Query Response.

3.1.2 Time Complexity and Space Complexity of KPS

Time Complexity:

The complexity of an algorithm is the cost of using the algorithm. Space complexity means how much memory is being utilized by an algorithm at run time. Time complexity is an amount of computation time needed to run an algorithm as in [32]. Let, the query in natural language consist of n number of unique tokens and m selected token which are already define in to the Grammar table. So the query consists of (m + n) number of tokens in total.

1	**2**	**3**	**.**	**.**	**n-2**	**n-1**	**n**	**1**	**2**	**3**	**.**	**.**	**m-2**	**m -1**	**m**

n tokens (Unique tokens) m tokens (Selected tokens)

In order to make a relation among unique tokens, it is required to have 3 tokens at first and then again these 3 tokens must be permuted together.

Equation for the unique tokens:

$$f(n) = ({}^{n}p_3) + (n \times {}^{3}p_3) = \frac{n!}{3!} + (n \times \frac{3!}{3!})$$

$$= \frac{n \times (n-1) \times (n-2)\ldots\ldots 4\times3\times2\times1}{3\times2\times1} + n$$

$= [\{n\times (n\text{-}1) \times (n\text{-}2)\ldots\ldots4\} + n] \rightarrow$ (Expression- 1)

From this above expression it can be divided in two parts.

$\{n\times (n\text{-}1) \times (n\text{-}2)\ldots\ldots\ldots4\} \rightarrow 1^{st}$ half.

$n \rightarrow 2^{nd}$ half.
We have to determine the numbers of terms in the first half of the above expression i.e. $\{n\times (n\text{-}1) \times (n\text{-}2)\ldots\ldots\ldots4\}$.
Suppose there are p numbers of terms in the first half of this equation.
Then t_p from the above equation will be:- $t_p = 4$

[As the expression is product of terms which are in arithmetic progression]

$\Rightarrow$ a + (p-1) d = 4 [Here a = 1^{st} term of the expression, d = common difference.]

$\Rightarrow$ n + (p-1) (-1) = 4

$\Rightarrow$ n – p + 1= 4

$\Rightarrow$ p = n +1- 4 = n-3

So,

n(n-1)(n-2)……….. 4

= $n^{n-3} \pm f_1(n)$ [Where $f_1(n)$ is rest of the remaining terms]

= O (n^{n-3}) [We can ignore rest part i.e. $f_1(n)$ as it consists the diminishing terms (Powers are diminishing)

for example:

if $f(n) = n^2 + 2n + 1$ then f(n) is $O(n^2)$]

$T_1(n) = O(n^{n-3})$

and for second half of the Expression –

$T_2(n) = O(n)$

Now, total time taken for the unique tokens.

$T(n) = T_1(n) + T_2(n)$

$T(n) = O(n^{n-3}) + O(n)$

$T(n) = O(n^{n-3} + n)$

There are m numbers of selected tokens in the query string.

So, here g(m) = m = O (m)

Total time taken for the selected tokens

T(m) = O (m)

Now the total calculation of time taken by the Algorithm is T(n, m) where n and m are same time unit.

$T(n, m) = T(n) + T(m) = O(n^{n-3} + n) + O(m) = O(n^{n-3} + n + m)$

Where $n \geq 3$ as if $n < 3$ then the expression in Big O notation is ambiguous to us. As per algorithm it is impossible to construct a valid relation without three tokens at least.

Example:

Query:- "What are the courses offered by the University?"

Unique Tokens (n) : courses, offered, University $\Rightarrow$ 3

Selected Tokens (m) : What, are, the, by, the $\Rightarrow$ 5

Now calculate the time taken by the algorithm is-

$T(n, m) = O(n^{n-3} + n + m) = 3^{n-3} + 3 + 5 = 3^{3-3} + 3 + 5$

= 1 + 3 + 5

= 9 time units (Time units may be in nano second, µ second, mili second).

Space Complexity:

Space Complexity of a program (for given input) is the number of elementary objects that program needs to store during its execution. This number is computed with respect to the size n of the input data.

Let a query consists of n tokens (unique) and m selected tokens (that go to the grammar table already defined) i.e. the query consists of (m + n) tokens in total.

Analysis:

1	**2**	**3**	**.**	**.**	**n-2**	**n-1**	**n**	**1**	**2**	**3**	**.**	**.**	**m-2**	**m-1**	**m**

$\underbrace{\qquad\qquad\qquad\qquad}_{\text{n tokens (Unique tokens)}}$ $\underbrace{\qquad\qquad\qquad\qquad}_{\text{m tokens (matched tokens)}}$

For Unique Tokens-

f(n) =	n	+	3	+	n. $^3p_3$
	No. of cells required for Unique tokens.		No. of cells required for 3 tokens at a time.		No. of cells required to permute the previous three tokens at a time out of n tokens.

$$
\begin{aligned}
f(n) &= n + 3 + n.\,{}^3p_3 \\
&= n + 3 + (n.3! / 3!) \\
&= n + 3 + n.1 \\
&= n + 3 + n \\
&= 2n + 3
\end{aligned}
$$

From the above expression the first term (2n) is of the highest power. So, as per Big O notation f (n) = O (n).

Therefore,

Space Complexity for the algorithm (for unique tokens) T_1 and an Input X.

$$DSPACE\ (T_1, X) = O\ (f\ (n)) = O\ (n)$$

For the Matched tokens-

$$g\ (m) = m = O\ (m)$$

Therefore,

$$DSPACE\ (T_2, X) = O\ (g\ (m)) = O\ (m)$$

Then

Total space taken by the algorithm

$$DSPACE\ (T_1, T_2, X) = O\ (n) + O\ (m) = O\ (n + m)$$

Example:

Query: - "What are the courses offered by the University?"

$$DSPACE\ (T_1, T_2, X) = O\ (n + m) = 3+5 = 8 \text{ Cells}$$

(Cell unit may be in bits, byte, kb, etc.)

3.2 GR based Automated Knowledge Provider System (AKPS):

GR based Automated Knowledge Provider System (AKPS) tool has been developed for proper extraction, utilization and dissemination of knowledge as in [4]. The AKPS is an automated and grammatical rule-based system. It does not contain any type of Knowledge Worker (KWs) concept. This system can easily map the natural language query to the physical data model of database.

3.2.1 Architecture:

The below architecture shows that GR based AKPS receives NL queries as a request from the Client side and sends response to the client side using the default database which is well defined and organized. If a query is not managed by the system, then unmanaged query will be stored into the temporary database and will wait for the update. The unmanaged query will be handled by the System Administrator, Database Administrator or any other Administrator. The management of unmanaged query has been discussed in [4]. When the update will complete, then GR based AKPS automatically will generate the response to the Client side. The System always will be updated by the many sources and amount of knowledge will be increased in this manner.

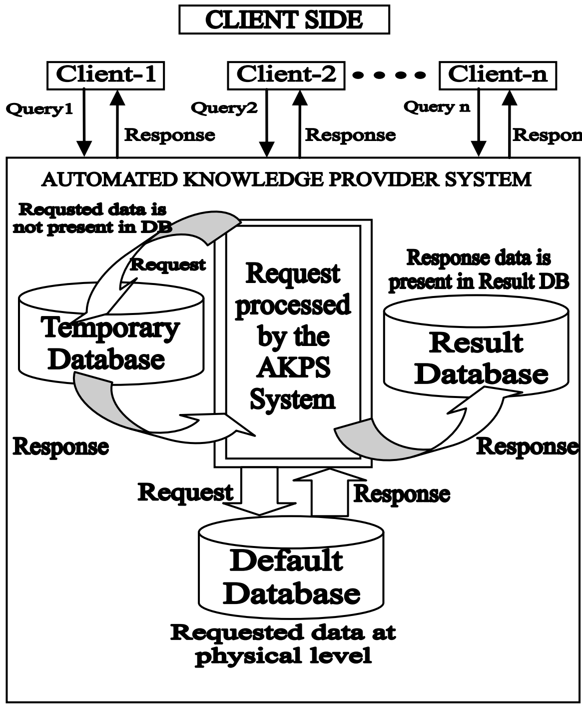

Fig.3: Architecture of AKP

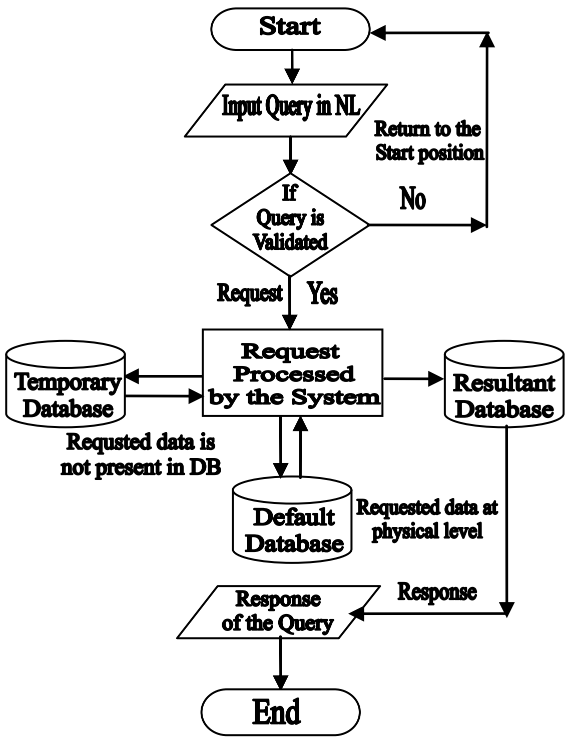

Fig. 4: Context diagram of AKPS

3.2.2 Algorithm and its Rule

The query will be posted from the client side in natural language and query sentence may be in Assertive or Interrogative. The parsing algorithms for Interrogative and Assertive statements are discussed in the following part of this section.

Interrogative Sentence formation:-

Interrogative Sentence - WH Phrase (attached Auxiliary Verb) + Noun Phrase +Verb Phrase + Complement.

WH phrase contain the Who, Which, What, etc. + am, is, are, were, was etc.

Table 1: Phrases of Interrogative Sentence

Sl no	Wh Phrase	Noun Phrase	Verb Phrase	Complement
1.	Wh	Noun	Verb	Noun
2.	Wh + Aux	Determiners +Noun		Preposition + Noun
3.				Preposition + Determiners + Noun

Assertive Sentence formation:-

Assertive Sentence - Noun phrase + Verb phrase + Complement.

Table 2: Phrases of Assertive Sentence

Sl no	Noun Phrase	Verb Phrase	Complement
1.	Noun	Verb	Noun
2.	Pronoun	Aux + Verb	Determiners + Noun
3.	Determiners + Noun	Aux + Adverb + Verb	Preposition + Noun
4.	Adjective + Noun	Preposition + Verb	Preposition + Determiners + Noun
5.	Determiners + Adjective + Noun		Preposition + Preposition + Adjective + Determiners + Noun

Both of these sentence formations have been used to create the grammatical rules. The grammatical rules are used to identify the Verb and Noun from query sentence. The verb may appear in the first position in an interrogative sentence or in the third position in an exclamatory sentence because there is no fixed position of the verb in a sentence in English Language. Verb is assumed to be the main connector of the entities as in [3]. Nouns are termed as an entities and verb makes a relationship between Nouns. Then system can do the formation of conceptual model, logical model and physical model from NL query.

Example: “What is the placement record of Calcutta University?”

Sentence formation of this example will be-

Wh Word + Auxiliary Verb + Determiner + Noun + Noun+ Preposition + Noun + Noun

Sometimes if a main verb is not present in a sentence then Auxiliary Verb completes the sentence in absence of main verb. In the above example main Verb is not present but Auxiliary Verb acts as a main Verb.

Rules:-

i. Pronouns = {I, We, He, She, They, Them, They} shall be termed “User”.

ii. Two Nouns and a Verb, or, Combination of Noun,
Pronoun and Verb may form ER- Diagram.

iii.If three or four or many more Noun may be recovered from the sentence then Noun+Noun or Noun+Noun+Noun will be treated as single Noun.

Algorithm:

1. Read Input Statement S.
2. Split Input Statement S into the String Array called STR[n].
3. Create a text file to use it as a log file. // enter value of Transaction ID, NLP Query ID, NLP Query into the log file in text format. Query Validation starts here.
   i. Declare Parts of Speech List:
   WH[n]={”what”,”which”…} // User NLP Query Validation.
   AUX[n]={“is”,”am”,”are”…}
   PRE[n]={“of”,”to”,”in”,”for”…}
   DET[n]={“a”,”an”,”the”…}
   PRO[n]={“I”,”you”,he”….}
   ii. Each value of STR[n] is checked with Parts of Speech List and counts the unique tokens using COUNT variable.
   IF COUNT < 2 THEN

Invalid Query Response sends to the Client
End.
// Insert value of NLP Query Validation Status "failed" into the log file in text format.
ELSE
// Insert value of NLP Query Validation Status "Passed" into the log file in text format.
First check the STR[0] equal to the words- "who","which","what"….etc.
IF STR[0]== "who"|| IF STR[0]== "which"|| IF STR[0]== "what" ….. THEN
// Insert text value "Query is Interrogative sentence" into the log file.
CALL Interrogative_Sentence(STR[n]) method
ELSE
// Insert text value "Query is Assertive sentence" into the log file.
CALL Assertive_Sentence(STR[n]) method
END IF

4. Interrogative_Sentence(STR[n]) method:
   i. Create the each Rule and store it on Rules Array.
      // Insert text value "Grammatical Rules created at runtime." into the log file.
   ii. Check the STR length with the each rule length from Rules[n].
      If the rule match from Rules[n] then send the selected rule and STR[n] to Parsing method which will either return 0 or 1 to a variable. // Insert text value "Grammatical Rules selected and sent with interrogative query sentence for Parsing" into the log file.
      If parsing method returns value 1 then
      // Insert text value "Interrogative Query Sentence parsed successfully" into the log file.
      Insert STR[n] in Sentence Table in Database and Insert selected Rules in Rules Table in Database.
5. Assertive_Sentence(STR[n]) method
   i. Create the each Rule and store it on Rules Array.
      // Insert text value "Grammatical Rules created at runtime." into the log file.
   ii. Check the STR. length with the each rule length from Rules[n].
      If the rule match from Rules[n] then send the selected rule and STR[n] to Parsing method which will either return 0 or 1 to a variable.
      // Insert text value "Grammatical Rules selected and sent with assertive query sentence for Parsing" into the log file.
      If parsing method returns value 1 then
      // Insert text value "Assertive Query Sentence parsed successfully" into the log file.
      Insert STR[n] in Sentence Table in Database and Insert selected Rules in Rules Table in Database.
6. int Parsing(string Rules, string STR)
   i. Declare Parts of Speech List:
      WH[n]={"what","which"…}
      AUX[n]={"is","am","are"…}
      PRE[n]={"of","to","in","for"…}
      DET[n]={"a","an","the"…}
      PRO[n]={"I","you",he"….}
      UNK[n]={"Noun","Adjective","Verb"….}
   ii. split Rules to R[n] and STR to S[n].
      Declare Result[0....n] = NULL.
      Check each word from S[n] with each word from WH[n], AUX[n], DET[n],PRO[n] and UNK[n] and store result in Result Array.
   iii. Initialize the variable value=0
      FOR i=0,j=0, TO Result.length -1 AND R.length-1 STEP 1
      IF Result[i] != R[j] THEN
      Value INCREMENTED BY 1
      END IF
      END FOR
      If Value > 0 then the parsing method will return 0 else 1.

7. Select Row one by one from Rules Table and Sentence Table in Database.
   i. Each Row of Rules Table and Sentence Table initialize in SS1 [n] and SS2 [n].
   ii. Each word from SS1 array inserts it into the grammar table as per SS2 array.

8. From Grammar Table select the values from Noun, Verb columns and properly initialize in Semantic Table.

3.2.3 Time Complexity and Space Complexity of AKPS

Time Complexity:
For each query, the System checks for Interrogative or Assertive sentence through its first token. In Next step, the query will go through some predefined rules to match and a rule will be selected when length of a rule is same as the length of a query before entering into parsing. Corresponding to each selected rule and a query, at a time, each token will go

through each of 6 grammar table array (Where each table array consists on n number of predefined words in the parsing method). One token will definitely match with one word of each of 6 grammar table array.

Let,

m = Number of matched tokens in a query.

p = Number of non matched tokens in a query.

s = Each Predefined grammar table array in the parsing.

The Function is derived as per the parsing algorithm of AKPS –

f(m,p,s) =	(1 + 5s)m	+	p(6s)	+	(m + p)
	For matched Tokens in a query.		For Non-matched tokens In a query.		Sum of number of matched tokens and non matched tokens.

Where, (1+5s)m represents matched tokens in a query. There are 6 grammar table arrays, each array denoted by s. A word is compared with words in each of the 6 table arrays, wherein 1 in (1+5s) corresponds to the event where the match for the word is found. 5s represents the matching effort in other 5 table arrays. (1+5s)m corresponds to the matching of m such words.

While checking for the match of query words in the table arrays, p(6s) corresponds to the unmatched words.

p corresponds to the number of non-matched tokens and each token from p tokens will search for match in all six grammar table arrays. So, for a single non matched token, it will be 6s and for p non-matched tokens, it will be p(6s).

(m+p) represents the query string in iteration when a rule is being selected as per equality of length of query string and rule.

Therefore,

$f(m,p,s) = (1 + 5s)m + p(6s) + m + p = m + 5sm + 6ps + m + p$

Or,

$f(n) \cong s + 5.n.n + 6.n.n + n + n = 3n + 5n^2 + 6n^2 = 11n^2 + 3n = O(n^2)$

[≅ means congruence i.e. its compatibility with the structure. Here, $m \cong p \cong s \cong n$ as either they are tokens or words.]

Time taken by the Algorithm- $T(n) = O(n^2) + 1$

Example:

Query: - "What are the courses offered by the University?"

$T(n) = O(n^2) + 1$

$= 8^2 + 1 = 65$ time units (Time units may be in nano second, µ second, mili second).

Space Complexity:

$f(m, p, r, s, l) = (m + p) + r + s + l + (m + p)$

where

(m + p) -> No. of total cells occupied by the query (where, m= No. of matched tokens in the query).

r -> No. of Rules definition Array.

s -> No. of Rules.

l -> Total No. of parts of speech lists where each list consist of n words in parsing function.

(m + p) = Length of newly created rules in the parsing function to match with the previously created rules.

Or,

$f(m, p, r, s, l) = (m + p) + r + s + l + (m + p)$

$\Rightarrow f(n) \cong n + n+n+n+6n+n+n = 12n = O(n)$ [≅ - congruence, Here, $m \cong p \cong r \cong s \cong l \cong n$.]

Therefore,

DSPACE (T, X) = O (f (n)) = O (n)

Example:

Query: - "What are the courses offered by the University?"

DSPACE (T, X) = O (n)

= 8 Cells (Cell unit may be in bits, byte, kb, etc.)

4 Time Complexity Graph of KPS and AKPS

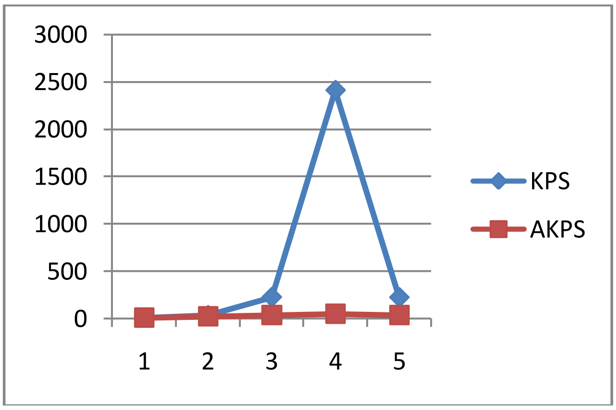

Fig.5: Time Complexity Graph

The blue colored line corresponds to time complexity of KPS and orange color line corresponds to AKPS in Fig: 5. Five unique queries have been considered for each of the KPS and AKPS systems. The equation of PC based KPS for the Time Complexity is O (n^{n-3} + n + m). The blue colored line shows that time has been taken by the KPS system for each query likewise 9,33,226,2411,223 Time Units (Time units may be in nano second, µ second, milli second). The Time Complexity equation of AKPS system is O (n^2) +1. The Orange colored line shows likewise 10, 26,37,50,37 Time Units (Time units may be in nano second, µ second, milli second) for the five queries. The Graph clearly shows that KPS system has taken more time other than AKPS system to process a query.

Table 3 Execution Time Units for KPS and AKPS.

Query	No. of unique tokens	No. of matched tokens	Time units (KPS)	Time Units (AKPS)
Q1	3	5	9	10
Q2	5	3	33	26
Q3	6	4	226	37
Q4	7	3	2411	50
Q5	6	1	223	37

The performance of the KPS system degrades as the number of unique tokens increase as per Table 3. Time Units calculations as per equation of KPS for 2 examples are shown below:

i. Execution Time for query- Q1:
(n^{n-3} + n + m)= 1+3+5 = 9 Time Units (Time units may be in nano second, µ second, mili second).

ii. Execution Time for query- Q4:
(n^{n-3} + n + m) = (7x7x7x7) +3+5 = 2411 Time Units (Time units may be in nano second, µ second, mili second).

When in AKPS takes only 10 Times Unit for query-Q1 and 50 Times Unit for query- Q4 as per the equation n^2 + 1.
Show calculations of AKPS:

i. Execution Time for query- Q1:
(n^2 + 1) = 9+1 = 10 Time Units (Time units may be in nano second, µ second, mili second).

ii. Execution Time for query- Q4:
(n^2 + 1) = $(7)^2$ + 1 = 49 + 1= 50 Time Units (Time units may be in nano second, µ second, mili second).

KPS system may be used to handle simple queries with few tokens and AKPS system is used to handle the complex type queries with excess tokens. Simple queries with few tokens shall not go for AKPS as there will be performance overhead. This is a good advantage for the AKPS system but it is true that the KPS system is not able to handle complicated queries with larger tokens where as AKPS system can handle more complicated queries easily. The AKPS system is rules based system and more rules can be implemented which will increase the capacity of handling of more complex type queries.

5 Space Complexity Graph of KPS and AKPS

The Blue colored bar corresponds KPS system and orange colored bar corresponds to AKPS system in Space Complexity Graph of Fig. 6. The equation of the KPS system is O (n + m) and AKPS system is O (n). Five unique queries have been utilized for the both systems but as per these equations, both the systems generally show same amount of space complexity in run time. The KPS system has taken 8 Cells (Cell unit may be in bits, byte, kb, etc.) as well as AKPS system has taken 8 Cells (Cell unit may be in bits, byte, kb, etc.) in run time for first query and remaining space has been taken by the both systems are 6, 4, 7, 4 Cells which is viewed in the graph for rest of the queries. This is a very good advantage for the AKPS system that it takes same amount of space where KPS system has taken same amount of space at run time. The AKPS system can able to handle more complicated queries in automated way using same amounts of space like KPS.

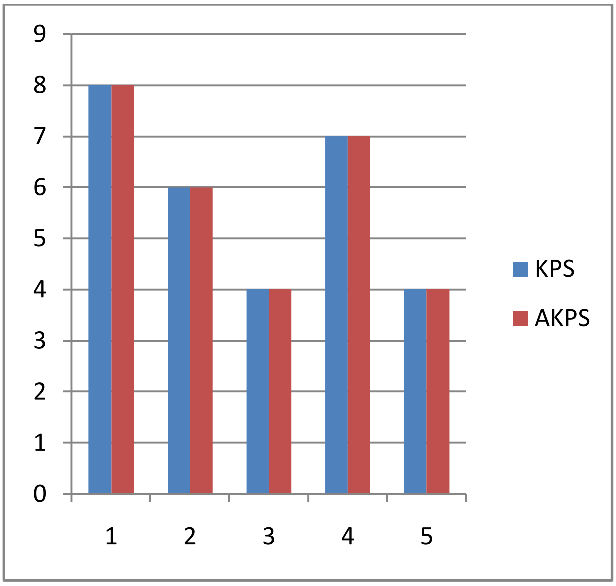

Fig. 6 Space Complexity Graph

6 KPS system is a subset of AKPS system

Optimization is a process where Maximizing and Minimizing a real function to find best values in a systematic way. There are many way to find best values from the real function in Optimization. As earlier in this paper the function T (n) of KPS system and T (n, m) of AKPS system have been created. The equations from T (n) and T (n, m) are below -

$T(n) = (n^{n-3} + n + m)$ form KPS system
and
$T(n, m) = (n^2 + 1)$ from AKPS system.
Initialized both of these equation in f (n) for KPS system and g (n) for AKPS system to continue the mathematical prove.
Let,
$f(n) = T(n)$ and $g(n) = T(n, m)$ then

i. f (n) function of KPS system - $f(n) = n^{n-3} + n + m$

ii. g (n) function of AKPS system - $g(n) = (n^2 + 1)$

Next to calculate the Maximum and Minimum value of function of f (n) and g (n).

6.1 Maximum (Maxima) and Minimum (Minima) value of function f (n) of KPS system

To calculate the Maximum and Minimum value of function if f(x) be a maximum, or a minimum at x=c, and if f ' (c) exists, then f ' (c)=0.

By definition, f(x) is a maximum at x=c, and find a positive number δ such that

$$f(c + h) - f(c) < 0 \text{ whenever } -\delta < h < \delta, (h \neq 0).$$

Therefore

$\frac{f(c+h)-f(c)}{h} < 0$ if h be positive and sufficiently small, > 0
If h be negative and numerically sufficiently small.

Thus, $\mathrm{Lt}_{h\to 0+} \frac{f(c+h)-f(c)}{h} \leq 0$, - See Ex. 5, 3.11 in [33].

and, similarly, $\mathrm{Lt}_{h\to 0-} \frac{f(c+h)-f(c)}{h} \geq 0$.
if f ' (c) exists, the above two limits, which represents the right-hand and left hand derivatives respectively of f(x) at x=c, must be equal. Hence, the only common value of the limit is zero. Thus, f ' (c) = 0 as in [33].

$f(n) = n^{n-3} + n + m \Rightarrow f(n) \cong n^{n-3} + n + n$ [≅ - congruence, here $m \cong n$.]

$\Rightarrow f'(n) = \frac{d}{dn}(n^{n-3}) + \frac{d}{dn}(2n) = \frac{d}{dn}[h(n)] + 2 = h'(n) + 2$ ….. (equation -1)

Now calculate the value of h (n),

$h(n) = n^{n-3} \Rightarrow \log h(n) = \log n^{n-3} \Rightarrow \log h(n) = (n-3)\log n \Rightarrow \frac{d}{dn}\log h(n) = \frac{d}{dn}(n-3)\log n$

$\Rightarrow \frac{1}{h(n)} h'(n) = \frac{n-3}{n} + \log n$ ……(equation - 2)

$\Rightarrow h'(n) = h(n)\left[\frac{n-3}{n} + \log n\right] = n^{n-3}\left[\frac{n-3}{n} + \log n\right] \simeq \left[\frac{n-3}{n} + \log n\right]$

from equation – 1

$f'(n) \simeq \frac{n-3}{n} + \log n + 2$

$f''(n) \simeq \frac{1}{n}\frac{d}{dn}(n-3) + (n-3)\frac{d}{dn}\left(\frac{1}{n}\right) + \frac{d}{dn}\log n + \frac{d}{dn}2 = \frac{1}{n}(1-0) + (n-3)\left(-\frac{1}{n^2}\right) + \frac{1}{n} + 0$

$= \frac{1}{n} - \frac{n-3}{n^2} + \frac{1}{n} = \frac{2}{n} - \frac{n-3}{n^2}$

For Maximum and Minimum value of f (n)

f ' (n) = 0

$\Rightarrow \frac{n-3}{n} + \log(n) + 2 = 0 \Rightarrow \frac{n-3+n\log(n)+2n}{n} = 0 \Rightarrow 3n-3+n\log(n) = 0$

Put the value of log (n).

[Therefore $\log(1+n) = n - \frac{n^2}{2} + \frac{n^3}{3} - \frac{n^4}{4} + \ldots\ldots..0$

Putting n = n − 1

$\Rightarrow \log(n-1+1) = (n-1) - \frac{(n-1)^2}{2} + \frac{(n-1)^3}{3} - \ldots\ldots\ \infty \quad \Rightarrow \log(n) = (n-1) - \frac{(n-1)^2}{2} + \frac{(n-1)^3}{3} - \ldots\ldots\infty$

$= (n-1)$

as the higher terms are ignored because of their small values]

$\Rightarrow 3n-3+n(n-1) = 0 \quad \Rightarrow 3n-3+n^2-n = 0$
$\Rightarrow n^2+2n-3 = 0 \quad \Rightarrow n^2+3n-n-3 = 0$

$\Rightarrow n(n+3)-1(n+3) = 0 \quad \Rightarrow (n+3)(n-1) = 0$

either $n+3 = 0 \Rightarrow n = -3$ or $n-1 = 0 \Rightarrow n = 1$

Now,

$f''(n) = \frac{2}{n} - \frac{n-3}{n^2}$

$f''(-3) = \frac{2}{-3} - \frac{-3-3}{(-3)^2} = -\frac{2}{3} + \frac{6}{9} = -\frac{2}{3} + \frac{2}{3} = 0$

$f''(1) = \frac{2}{1} - \frac{1-3}{1^2} = 2 + \frac{2}{1} = 4 > 0$ (Minimum Value)

at n = 1, f (n) has minimum value and

$f_{min}(n) = n^{n-3} + n + n = 1^{1-3} + 1 + 1 = 1^{-2} + 2 = \frac{1}{1^2} + 2 = 1 + 2 = 3$ the domain of f (n) is [3, ∞) .

6.2 Maximum (Maxima) and Minimum (Minima) value of function g (n) of AKPS system

$g(n) = n^2 + 1 \quad \Rightarrow \frac{d}{dn}n^2 + \frac{d}{dn}1 \quad \Rightarrow g'(n) = 2n + 0 = 2n \quad \Rightarrow \frac{d}{dn}g'(n) = \frac{d}{dn}2n \quad \Rightarrow g''(1) = 2$

for maximum and minimum value of g (n) :

$g'(n) = 0 \Rightarrow 2n = 0 \Rightarrow n = 0$

Now,

$g''(n) = 2$ put the value of n,

$g''(0) = 2 > 0$ (minimum value) and $g(n) = 0^2 + 1 = 1$ the domain of g (n) is [1, ∞) .

6.3 Graph Representation of Minima and Maxima values of function f (n) and g (n)

The function of KPS system f (n) is [3, ∞) and function of AKPS system g (n) is [1, ∞). Below the graph shows both the values of f (n) and g (n) function-

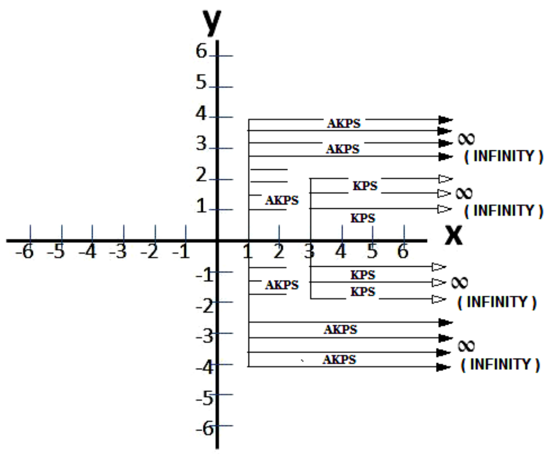

Fig.7 Graph Representation of f (n) and g (n) function.

The function f (n) of KPS system has minimum value 3 and graph (Figure7) shows that line has been started from 3 in X- axis. The minimum value can be 4, 5, 6 or any other value but not less than 3. In other side the maximum value will be infinity. The maximum value does not have any limit.
The function g (n) of AKPS system has minimum value 1 and maximum value is infinity as same as the function f (n) of KPS. In graph the line has started from points 1 and this point is the minimum starting point of AKPS in X- axis. The starting point

means minimum value of AKPS system in X- axis. The minimum value can be 2, 3,4,5,6 or any other value but not less than 1. The lines of AKPS in the graph are moving to the maximum value of function g (n) which is infinity. It is possible that AKPS system may be started from point 3 in X-axis and it is also the minimum point of KPS system. But KPS system cannot be started from point 1 in X-axis where the minimum value of AKPS system is 1. It is proved mathematically that AKPS has wider acceptance (in the sense of mathematical optimization) than KPS. As per the graph KPS system is the subset of AKPS system.

KPS ⊆ AKPS

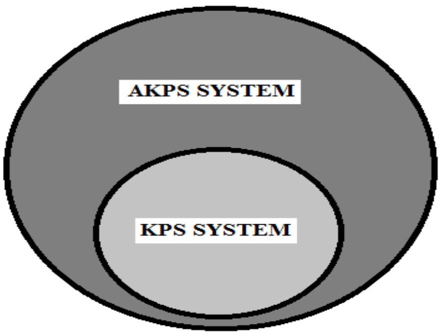

Fig. 8 Euler diagram of KPS and AKPS

Here the Euler diagram showing KPS system is a proper subset of AKPS system and conversely AKPS system is a proper superset of KPS system.

7 Conclusion and Future Work

The proposed KPS is a basic system of the Knowledge Management System. It handles natural language query as a request and after processing it generates the response to the client side. It has many applications on e-services like e-Tourism, e-Hospitality management, e-Banking and others. The KPS system can handle interrogative, assertive sentences as a query. The main act of this KPS system is to generate the Verb(s) and Noun(s) from the sentence(s). Verb makes the relationships between Nouns and represents E-R Diagram of database. But if a Verb not present in the sentence then KPS system does not have the ability to create the relationship between Nouns using Auxiliary Verb. This affects the entire system because right sentence may be rejected because of absence of a main verb. Many times two or three unique words indicate a single Noun but KPS system does not recognize it. Then this situation cannot be handled by the KPS system and in result the KPS system failed to generate the query-response. This KPS system is not an automated one and mostly dependent on the Knowledge Workers (KWs). The AKPS system has been developed to manage all these draw backs in KPS system. The AKPS system is an automated system and Knowledge Workers (KWs) concept has been removed. The AKPS system follows the grammar rules and creates the E-R diagram using grammar rules. The AKPS system works on grammar rules and rules may be increased for handling interrogative and assertive sentences. The AKPS system is updated version of KPS system. The future work of this AKPS system is to handle voice query as well as text queries. It has been proved that KPS system is a subset of AKPS system and both systems together can be applied in many application areas. KPS system is able to handle simple query in assertive and interrogative very fast as well as AKPS system can handle complex type queries in assertive and interrogative. Future work emphasizes on developing a hybrid KPS which may operate on either mode of KPS and AKPS which an option to switch over to either mode depending on tokens generated from a user query.

References:

[1] Baisakhi Chakraborty, Dr. D.Ghosh, Narayan C. Debnath, “A Knowledge Worker Centric KMS for Smart e-services”, *Journal of Computational Methods in Sciences and Engineering*, IOS Press, Amsterdam, Netherlands, Vol.11, Issue-3, August 2011, pp. 41-54.

[2] A.H.M. Majid, S.P. Lee, and S. Salwah, “An Ontology-based Knowledge Model for Software Experience Management”, *International Journal of the Computer, the Internet and Management*. Vol. 14, No.3, 2006, pp. 79--88.

[3] P. Mukherjee, B. Chakraborty, “Knowledge Provider System with natural language data modeling”, *ISCA 29th International Conference on Computers and their Applications*, Las Vegas, USA, 2014, pp. 229-236.

[4] P. Mukherjee, B. Chakraborty, “Automated Knowledge Provider System with Natural Language Processing”, *Accepted for publication in Special Issue: Frontiers in Intelligent Systems Development, Institute of Integrative Omics and Applied Biotechnology (IIOAB) Journal*, 2015.

[5] J. Li, "Sharing Knowledge and Creating Knowledge in Organizations: The Modelling, Implementation, Discussion and Recommendations of Weblog-based Knowledge Management", Proceedings of *IEEE, International Conference on Service Systems and Service Management*, 2007, pp. 1-6.

[6] Z. Deng, and X. Hu, "Discussion on Models and Services of Knowledge Management System", Proceedings of *IEEE ISITAE* '07, 2007, pp. 114--118..

[7] Sonal Gupta and Christopher D. Manning, "Improved Pattern Learning for Bootstrapped Entity Extraction", In Proceedings of the *Eighteenth Conference on Computational Natural Language Learning (CoNLL)*, 2014.

[8] Prathvi Kumari, Ravishankar K, "Measuring Semantic Similarity between Words using Page-Count and Pattern Clustering Methods", *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* Vol. 3, Issue-2, July 2013, pp. 31-34.

[9] Reinaldha, Filbert , Widagdo, Tricya E., "Natural Language Interfaces to Database (NLIDB):Question Handling and Unit Conversion", *International Conference on Data and Software Engineering (ICODSE),* 2014, pp. 1 – 6.

[10] Uduwela, C. Wasana, Gamini Wijayarathna, "Survey on Tools and Systems to Generate ER Diagram from System Requirement Specification", *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2014, pp. 370 – 373.

[11] Yaru Wang ; Aijun Li ; Yuan Jia," Stress distribution based on dependency par sing ofChinese discourse", *17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA),* 2014, pp. 1 – 5.

[12] Park, Jihyoun, "Integrated Visual Analytics Tool for Heterogeneous Text Data", *IEEE Conference on Visual Analytics Science and Technology (VAST),* 2014, pp. 325-326.

[13] S. Arivarasan, "Improving Habitability of NLI for Querying Ontologies with Feedback", *International Conference on Information Communication and Embedded Systems (ICICES),* 2014, pp. 1 – 6.

[14] A. Agarwal, S. Mulpura, R.S. Behara,V. Tyagi, "Domain Independent Natural Language Processing –A Case Study for Hospital Readmission with COPD", *IEEE 14th International Conference on Bioinformatics and Bioengineering*, 2014 pp. 399 – 404.

[15] Baisakhi Chakraborty, Meghbartma Gautam, "A Personnel Centric Knowledge Management System", *Contemporary Computing Communications in Computer and Information Science,* Volume 40, 2009, pp. 263-272.

[16] Smita Selot, Neeta Trpathi, A.S Zadgaonkar, "Knowledge Representation in *pAninI* Framework Using Neural Network Model", *BIJIT - BVICAM's International Journal of Information Technology,* Vol. 5, No. 1, 2013, pp. 537-544.

[17] K. Englmeier, J. Pereira, and J. Mothe, "Choreography of web services based on natural language storybooks", *In Proceedings of the 8th International Conference on Electronic Commerce*, 2006, pp. 132-138.

[18] L.A.E. Al-Safadi, "Natural Language Processing for Conceptual Modeling", *International Journal of Digital Content Technology and its Applications*, Vol 3(3), 2009, pp. 47-59.

[19] Przemysław Maciołek, Grzegorz Dobrowolski, "CLUO: WEB-SCALE TEXT MINING SYSTEM FOR OPEN SOURCE INTELLIGENCE PURPOSES", *biblioteka główna*, Vol. 14 No.1, 2013, pp. 45-62.

[20] Ana Patricia T. Alfonso, Illuminada Vivien R. Domingo, Mary Joy F. Galope, Ria A. Sagum, Rachelle B. Villar, and Jobert T. Villegas , "Named Entity Recognizer for Filipino Text Using Conditional Random Field", *International Journal of Future Computer and Communication*, Vol. 2, No. 5, October 2013, pp. 376-379.

[21] Steven K.C. Lo, "Modeling Knowledge Flow and Learning Strategy in Multi-agent System", *Information Technology Journal,* 2013, pp. 1717-1726.

[22] Bidyut Das, Subhajit Pal, Suman Kr. Mondal, Dipankar Dalui, Saikat Kumar Shome, "Automatic Keyword Extraction From Any Text Document Using N-gram Rigid Collocation", *International Journal of Soft Computing and Engineering (IJSCE),* Volume-3, Issue-2, 2013.

[23] Wafa N Bdour1 , Natheer K. Gharaibeh, “Development of Yes/No Arabic Question Answering System*”, International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol.4, No.1, January 2013, pp. 51-63.

[24] Adel Tahri, Okba Tibermacine,“DBPEDIA BASED FACTOID QUESTION ANSWERING SYSTEM”, *International Journal of Web & Semantic Technology (IJWesT)*, Vol.4, No.3, July 2013, pp. 23-38.

[25] F.S.C. Tsengb, A.L.P.Chen and W.P. Yang, “On mapping natural language constructs into relational algebra through E-R representation”, *Data and Knowledge Engineering*, Elsevier, Vol. 9, Issue 1, 1992, pp. 97-118.

[26] T. Catarci, M. F Costabile, S.Levialdi and C.Batini,“Visual Query Systems for Databases: A Survey”, *Journal of Visual Languages and Computing*, 1997, pp. 1.

[27] I. Androutsopoulos, G.D. Ritchie, P. Thanisch, “Natural Language Interfaces to Databases – An Introduction”, *Journal of Natural Language Engineering*, Cambridge University Press, Research Paper no. 709, 1994, pp. 15.

[28] Athira P. M., Sreeja M., P. C. Reghuraj, “Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System”, *International Journal of Web & Semantic Technology (IJWesT)*, Vol.4, No.4, October 2013, pp. 31-39.

[29] Alok Ranjan Pal, Diganta Saha, “Detection of Slang Words in e-Data using semi-Supervised Learning”, *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol. 4, No. 5, September 2013, pp. 49-60.

[30] Yinon Bentor Amelia Harrison Shruti Bhosale Raymond Mooney, “Slot Filling System: Bayesian Logic Programs for Textual Inference”, *University of Texas at Austin KBP*, 2013.

[31] Julia Neidert, Sebastian Schuster, Spence Green, Kenneth Heafield, and Christopher D. Manning, “Stanford University’s Submissions to the WMT 2014 Translation Task”, *In Proceedings of the ACL 2014 Ninth Workshop on Statistical Machine Translation*, 2014.

[32] Adam Duffy and Tom Dowling, “Algorithm Analysis- A Technical Report”, Version 1.4, Department of Computer Science, National University of Ireland, Maynooth, Maynooth, Co. Kildare, Ireland, November 13, 2003, pp. 5.

[33] B.C. Das, M.sc, B.N. Mukherjee,M.A, “Differential Calculas”, Revised 51st Edition, 2009, U.N. DHUR & SONS PRIVATE LTD.,Kolkata- 700073, 2009, pp. 316-317.