PTQL to SQL Query Transformation Using Incremental Information Extraction and RDBMS

Vrushali Patil
Department of Computer Engineerng,
PG student,
R.C.Patel College of Engineering,
patl.vrushali222@gmail.com

Prof. R. B. Wagh
Department of Computer Engineering,
Faculty of Computer Engineering,
R.C.Patel college of Engineering,
raj_wagh@rediffmail.com

Abstract:- Information Extraction, Database Query, PTQL, PTDB.

Abstract:- In the process of information extraction, it expressed in the form of database query. Information extraction used for unstructured or semi-structured machine readable document. and provide automated query generation components. Natural processing information implemented information extraction system. To perform extraction, the user does not require actual knowledge of query language, active approach in Information extraction process offers query generation components which are automated. Extraction result quality and efficiency are the major aspects of Information Extraction. We offer robust parsing algorithm majorly based on link grammar formalism and which is for parsing natural language. Information Extraction composed of two phase: Initial Phase and extraction phase.

*****

1. INTRODUCTION

Information extraction developed specific purpose program which contains sentence splitters, tokenizes, named entity recognizers, shallow or deep syntactic parsers. And the extraction based on a collection of patterns. Information extraction has elevated demand which results in the developments of framework such as UIMA[2] and GATE providing a way to perform extraction by defining workflows of components. This type of extraction is usually file based and processed data can be utilized between components. UIMA are powerful search capabilities and a data driven framework for the development, composition and distributed development of analysis engines.

UIMA technology mainly concentrates on the natural language processing. Researchers are occupied in activity ranging from natural language dialog, information retrieval, topic tracking, named entity detection, document classification and machine translation. UIMA provide four components [2]:

1. Acquisition
2. Analysis of unstructured information
3. Analysis structured information
4. Discovery of components

Information Extraction mainly has two phases:

Initial Phase: Text processing

Extraction Phase: Database query processing

Information extraction involves different type of text processing modules in order to execute relationship extraction.

Such extraction includes:

Sentence splitting: Identifies sentences from a paragraph of text

Tokenization: Recognition of word tokens in the sentence

Named entity recognition: Recognition of entity type of interest

Syntactic parsing: Recognition of grammatical structure of sentence

Pattern Matching: Bunch of extraction pattern that employ lexical, syntactic and semantic feature.

1.1 Link Grammar

Link grammar consists of two basic parameter directionality and distance. Link grammar is similar to dependency grammar includes a head dependant relationship.

Link grammar employed for information extraction of biomedical text and it is a dependency parser which based on link grammar theory. Also it is capable of outputting constituent tree [8].

Constituent tree is syntactic tree of a sentence with node represented by language and word of the sentence in leaf node. Link grammar is nothing but linking requirement between two words.

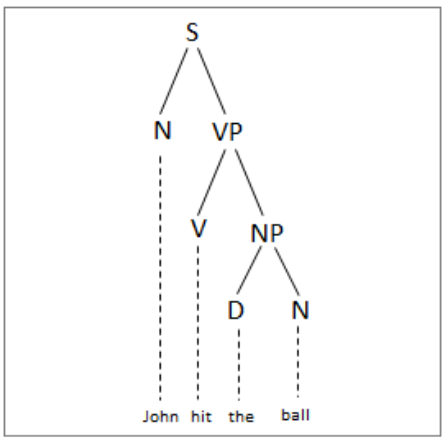

Figure1: Constituent based parse tree

 ISSN: 2321-8169
342– 345

Figure 1 shows the structure of parse tree starting from s and ending in each of the leaf node.

Figure1 consist of some abbreviation as follow:

S: For sentence

NP: Noun phrase

VP: Verb phrase, which serves as predicate

V: For verb phrase, which serves as transitive verb

N: for noun phrase.

1.2 Information Extraction

From the unstructured or semi-structured machine readable documents, information extraction automatically fetches required information in structured format. Because of the obscurity of problem, current approach to information extraction concentrate on restricted domains. Major goal of information extraction consist of allowing computation to be operated on unstructured data and also create simple machine readable text to process the sentence.

Classic subtask of information extraction includes:

Named Entity Recognition: Recognition involve a unique identifier to the extracted entity.

Co reference Resolution: It finds link previously extracted named entity.

Relationship Extraction: Identifies relationship between entities.

Information Extraction: Provide tools for building high performance, natural language application.

Table Extraction: finding and extracting of the tables from documents [1].

1.2.1 Initial Phase

Initial phase is used to generate parse tree and tagging of information in parse tree database.

1.2.2 Extraction Phase:

Parse tree query language is designed and implemented in this phase.

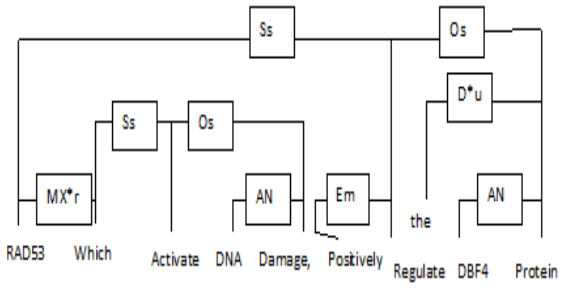

Figure 2: Shows Linkage of the sentence "RAD53, which activates DNA damage, positively regulates the DBF4 protein"

Figure2 describe tagger of information in information extraction.

2. EXISTING SYSTEM

Existing system consist of two phases: Initial phase which process the text and extraction phase extracts the data using database query. In the initial phase text processor is responsible for corpus processing and storage of process information in parse tree database. In the extraction phase PTQL query evaluator which receive PTQL query and transformed it into keyword based and SQL query. It will be evaluated by RDBMS and IR engine. Index builder creates an inverted index which set indexing to sentence and which brings speed in query evaluation process. Here we are converting PTQL query to SQL query transformation and developing standford dependency parser for parse the sentence.

2.1 Architecture of Our System

There are two approaches for generating PTQL query:

1. Training set driven query generation
2. Pseudo relevance feedback driven query generation

Training set driven query generation first automatically interprets an unlabeled document collection with information drawn from problem specific database. Even if the unavailability of training data, we use pseudo relevance approach to generate PTQL queries.

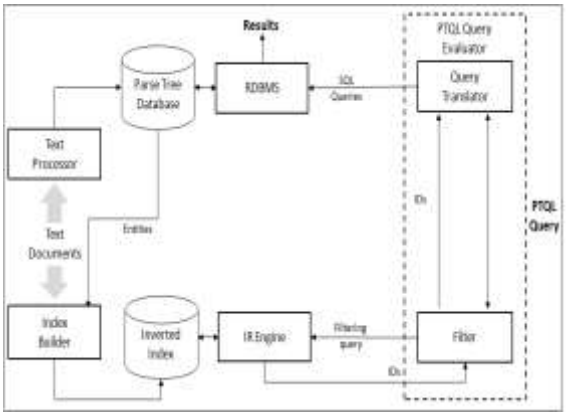

Figure 3: Architecture of PTQL Framework[1]

2.2 Parse Tree Query Language

Query language such as XPath[3],[6] and XQuery[12] are not suitable for linguistic pattern. We designed query language called parse tree query language. Extension of linguistic query language LPath is PTQL which allows queries to achieve constituent tree and linking between words on linkage.

A PTQL query consists of four modules:

1. Tree Patterns: Describe Hierarchical structure
2. Link Condition: Describe linking requirement between unlike nodes.
3. Proximity Conditions: Point out the words which are within specific number of words.
4. Return Expression: Returns value

 ISSN: 2321-8169
342– 345

2.3 Query Evaluation

Evaluation of PTQL query involve use of IR engine[4].
Following step use to evaluate PTQL query:

1. Translation of the PTQL query into a filtering query
2. Using the filtering query, it fetches the related documents and sentence from inverted index.
3. Translate PTQL query into SQL query.
4. Return the result as in form of SQL query

2.4 Query Generation

Generation of PTQL query involve PTQL query automatically. There are two approaches for generating PTQL query. Training set driven query generation and Pseudo-relevance feedback driven query generation.

2.4.1 Training Set driven Query Generation

PTQL queries are formed by query generator to perform extraction from the parse tree database. The generalized patterns are then translated into PTQL queries for extraction.

2.4.2 Pseudo Relevance Feedback Driven Query Generation

Pseudo-relevance feedback driven query generation automatically generate PTQL queries by considering the constituent trees of the top-k sentences retrieved with a Boolean keyword based query. We defined that two sentence are grammatically similar if they have the same mth level string encoding. The step of generating PTQL queries as follow [1]:

1. Fetches sentence from the inverted index and retrieve constituent tree of the top k sentence from PTDB.
2. For each sentence extract the subtree that is rooted at the lca(Least Common Ancestor) of all the query terms as leaf nodes from the constituent tree.
3. Generate M^{th} level string encoding for each of the subtree.
4. Sentence that are grammatically similar based on their M^{th} level string encoding are grouped together to form cluster.
5. PTQL query generated for each common grammatical patterns.

3. ALGORITHM

Here we describe robust parsing algorithm[7] for link grammar and develop parsing and pruning algorithm from unrestricted natural language. Goal of the robust parsing algorithm is to parse all sentence with minimum cost. In this case function takes as input indices of two words L and R, where words are numbered from 0 to n-1[8].

```
Parse
1.t<-0
2.for each disjunct d of word 0
3.        do if left[d]=NIL
4.                  then t<-t+COUNT(0,N,right[d],NIL)
5.return t

COUNT(L,R,l,r)
1.if R=L+1
2.     then if l=NIL and r=NIL
3.              then return l
4.               else return 0
5.else total<-0
6.      for W<-L+1 toR-1
7.              do for each disjunct d of word W
8.                    .....
```

Line 1-4 count procedure handle the boundry condition.

4. FINAL RESULT

In this paper we convert PTQL query to SQL query transformation using biocreative 3 IPS test data[5] and for extra modification we develop standford dependancy parser such as Pro3Gres and Stanford dependancy scheme, so that they can be stored in PTDB and query using PTQL.

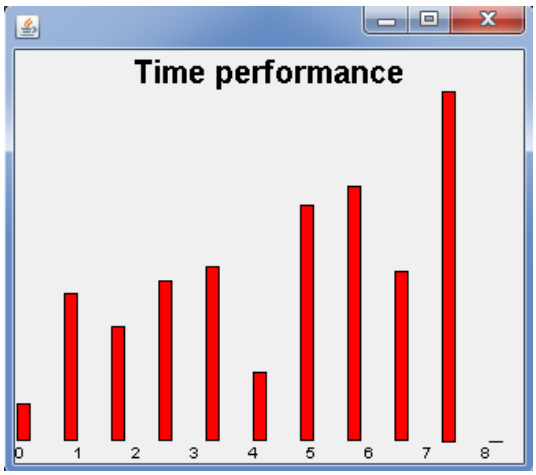

Figure 4: Time Performance of SQL Query

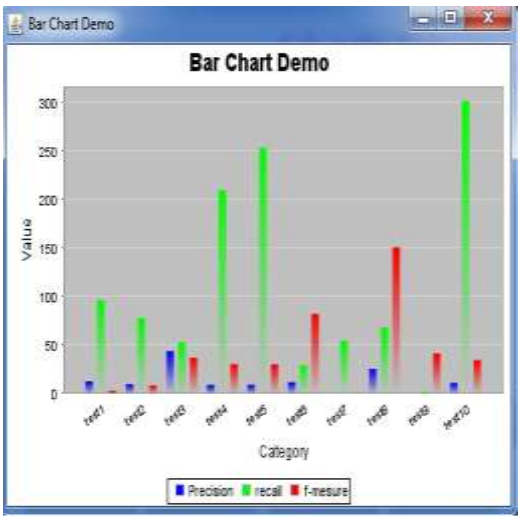

Figure 5: Time performance for PTQL evaluation

Table1: Performance pseudo relevance feedback query generation method or lung-cancer Metabolic Relations for SQL query

System	Precision	Recall	F-measure
Query(m=1)	0.76	8.3	228.32
Query(m=2)	28.52	5.5	77.66
Query(m=3)	0.73	1.03	85.7
Query(m=4)	6.51	26.17	3.70

5. COMPARISON

Here we develop nested query and part of sppech tagger of sentence.

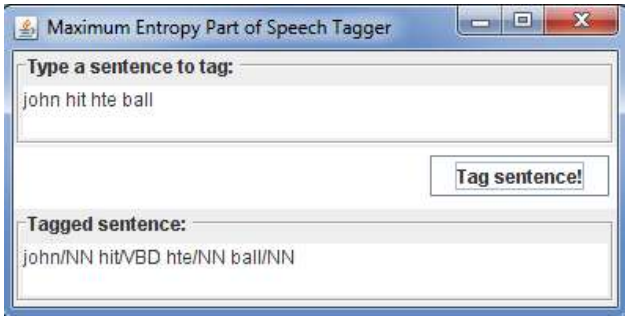

Figure 6: Tagger of Sentence

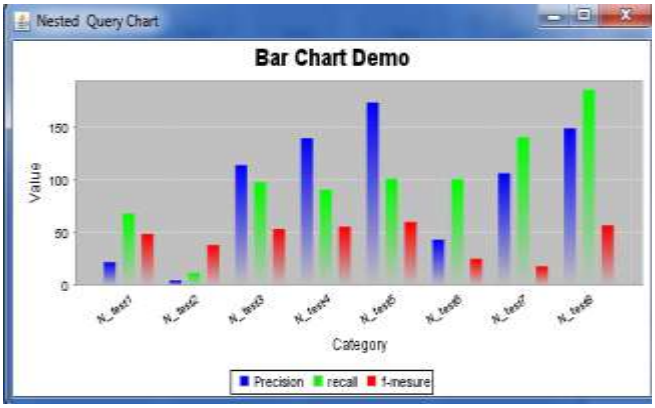

Figure 7: Subquery evaluation for the set of 8 queries against time performance.

Table2: Performance of pseudo relevance feedback driven query generation method or lung-cancer Metabolic Relations for Nested query

System	Precision	Recall	F-measure
Query(m=1)	27.16	31.73	24.88
Query(m=2)	2.86	109.01	64.49
Query(m=3)	7.67	61.49	101.3
Query(m=4)	35.04	40.94	32.1

6. REFERENCES

[1] Luis Tari, Jorg Hakenberg, Tran Cao Son, "Incremental Information Extraction Using Relational Database", *IEEE Trans. on knowledge and data engineering* , vol.24,No.1, pp. 86-99, January 2012.

[2] D. Ferrucci and A. Lally, "UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment," *Natural Language Eng.*, vol 10,nos. 3/4, pp.327-348, 2004.

[3] S. Bird et al.," Designing and evaluating an XPath Dialect for Linguistic Queries," in *Proc. 22th International Conf. Data Eng.(ICDE '06),*,2006.

[4] E. Agrichtein and L. Gravano, "Querying Text Databases for Efficient Information Extraction," in *Proc. Int'l Conf. Data Eng.(ICDE),* pp.113-124, 2003.

[5] W. Baumgartner, Z.Lu, H. Johnson, J. Caporaso, J. Paquette, E. White, O. Medvedeva, K. Cohen and L. Hunter, "An integrated Approach to Concept Recognition in Biomedical Text," *in Proc.Second Biocreative Challenge*,2006.

[6] S. Bird, Y. Chen, S.B. Davidson, H. Lee, an d Y. Zheng, "Extending XPATH to support Linguistic Queries," in Proc. Workshop Programming Language Technologies for XML (PLAN-X), 2005.

[7] D. Grinberg, J. Lafferty, and D. Sletor, "A Robust Parsing Algorithm for link Grammar," Technical Report CMU-CS-TR, pp.95-125, "Carnegie Mellon Univ.,1995.

[8] "XQuery 1.0: An XML Query Language," *http://www.w3.org/XML/Query,June2001.*