International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 10 | Oct -2017 www.irjet.net p-ISSN: 2395-0072

Automated Query Generation of Relational DBMS for Information and Knowledge Extraction

Surekha D. Naik[1], Vaishali Londhe[2], Nilima Nikam[3]

[1]*Student of ME dept. of Computer Engg, YTIET, Maharastra, India.*
[2]*Professor, dept. of Computer Engg, YTIET, Maharastra, India.*
[3]*Professor, dept dept. of Computer Engg, YTIET, Maharastra, India.*

Abstract - An Information extraction systems traditionally implemented as a pipeline of special-purpose processing modules targeting the extraction of a particular kind of information. A major drawback is, whenever a new extraction goal emerges or a module is improved, extraction has to be reapplied from scratch to the entire text corpus even though a small part of corpus might be affected. By using database queries, information extraction enables the generic extraction and minimizes re-processing of data. Furthermore, this provides automated query generation components so that, casual users no need to learn the query language in order to perform extraction. To demonstrate the feasibility of our incremental extraction approach, experiments can be performed to highlight two important aspects of an information extraction system: efficiency and quality of extraction results. The existing extraction frameworks do not provide the capabilities of managing intermediate processed data such as parse trees and information. It is also extended to per-sentence extraction, it is important to notice that the query language itself is capable of defining patterns across multiple sentences. Hence, in order to provide nearest and good result the incremental approach is compared with the existing systems.

Key words: Text Mining, query languages, information storage and information retrieval.

1. INTRODUCTION

Improving the ability of computer systems to process text is a significant research Challenge. IE has been an active research area that seeks techniques to uncover information from a large collection of text. Examples of common IE tasks include the identification of entities (such as protein names), extraction of relationships between entities (such as interactions between a pair of proteins) and extraction of entity attributes (such as co reference resolution that identifies variants of mentions corresponding to the same entity) from text. Many applications are based on partially structured databases, where structured data conforming to a schema is combined with free text. Information extraction systems traditionally implemented as a pipeline of special-purpose processing modules targeting the extraction of a particular kind of information. Most recent IE approaches are suitable for only static corpora.

1.1. Natural Language Processing

IBM Research has over 200 people working on Unstructured Information Management technologies with a strong focus on Natural Language Processing (NLP). These researchers are engaged in activities ranging from natural language dialog, information retrieval, topic-tracking, named-entity detection, document classification and machine translation to bioinformatics and open-domain question answering. An analysis of these activities strongly suggested that improving the organization's ability to quickly discover each other's results and rapidly combines different technologies and approaches would accelerate scientific advance. Furthermore, the ability to reuse and combine results through a common architecture and a robust software framework would accelerate the transfer of research results in NLP

1.2. Objectives and Scope

Information Extraction is a technology that is futuristic from the user's point of view in the current information-driven world. Rather than indicating which documents need to be read by a user, it extracts pieces of information that are salient to the user's needs. The idea behind information extraction is to make it possible to easily identify and assimilate data that is relevant to a particular activity, without the need to manually go through large amounts of information to find exact data required.

Document search and retrieval applications tailored to the needs of life science. Structure extraction is useful in a diverse set of applications. We list a representative subset of these, categorized along whether the applications are enterprise, personal, scientific, or Web-oriented, Enterprise Applications, News Tracking, Biomedical research, medical literature which contains information about new treatments for diseases. Similarly, news archives contain information useful to analysts tracking financial transactions, or to government agencies that monitor infectious disease outbreaks. etc.

2. LITERATURE SURVEY

Information extraction has been an active research area over the years. The main focus has been on improving

the accuracy of the extraction systems, and IE has been seen as a one-time execution process. Such paradigm is inadequate for real-world applications when IE is seen as long as running processes.

Rule-Based IE Approaches

In general, a document is broken up into chunks (e.g., sentences or paragraphs), and rules or patterns applied to identify entities. Different operations such as joins in RDBMS are performed over extracted facts that stored in various database tables. Rules are then applied to integrate different types of extracted facts. However, these rules are not capable of querying parse trees.

Machine Learning Approaches for IE

Machine learning a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. For example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders.

Information Extraction:

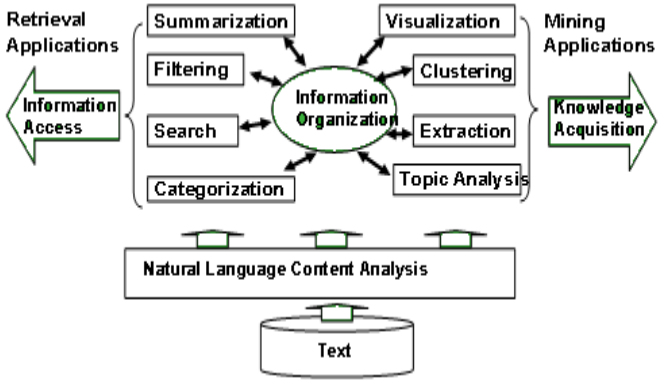

Fig: Conceptual Framework of Text Information System

METHODOLOGY

System Approach:

A Novel Database-Centric Framework for Information Extraction. We first give an overview of our approach, and discuss each of the major components of our system. Abstract— Information extraction systems traditionally implemented as a pipeline of special-purpose processing modules targeting the extraction of a particular kind of information. A major drawback is, whenever a new extraction goal emerges or a module is improved, extraction has to be reapplied from scratch to the entire text corpus even though a small part of corpus might be affected. By using database queries, information extraction enables the generic extraction and minimizes re-processing of data. Furthermore, this provides automated query generation components so that, casual users no need to learn the query language in order to perform extraction. To demonstrate the feasibility of our incremental extraction approach, experiments can be performed to highlight two important aspects of an information extraction system: efficiency and quality of extraction results. The existing extraction frameworks do not provide the capabilities of managing intermediate processed Data such as parse trees and information. It is also extended to per-sentence extraction, it is important to notice that the query language itself is capable of defining patterns across multiple sentences. Hence, in order to provide nearest and good result the incremental approach is compared with the existing Systems.

Our approach composed basic two phases:

- **Initial phase:** for processing of text and
- **Extraction phase:** for using database queries to perform extraction.

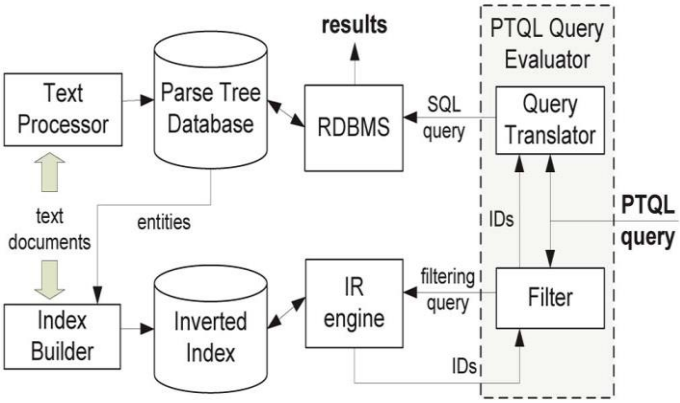

Fig System architecture of the PTQL framework

As shown in fig the Text Processor in the initial phase is responsible for corpus processing and storage of the processed information in the Parse Tree Database (PTDB). The extraction patterns over parse trees can be expressed in our proposed parse tree query language. In the extraction phase, the PTQL query evaluator takes a PTQL query and transforms it into keyword-based queries and SQL queries, which are evaluated by the underlying RDBMS and information retrieval (IR) engine. To speed up query evaluation, the index builder creates an inverted index for the indexing of sentences according to words and the corresponding entity types. Fig 5.1 illustrates the system architecture of our approach.

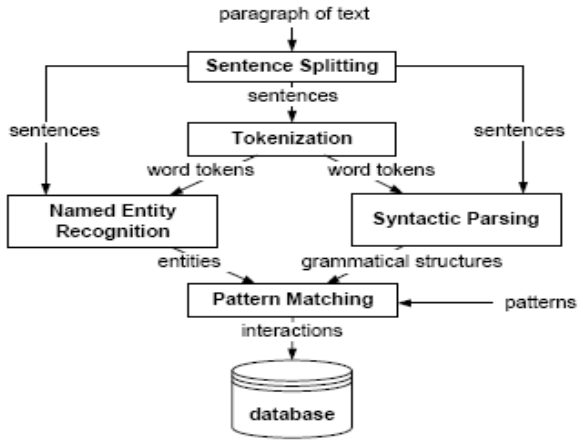

Fig.A workflow of text processing modules that takes a paragraph of text as input to perform relationship extraction

A typical IE setting involves a pipeline of text processing modules in order to perform relationship extraction. These include:

1. Sentence splitting: identifies sentences from a paragraph of text.

2. Tokenization: identifies word tokens from sentences.

3. Named entity recognition: identifies mentions of entity types of interest.

4. Syntactic parsing: identifies grammatical structures of sentences.

5. Pattern matching: obtains relationships based on a set of extraction patterns that utilize lexical, syntactic, and semantic features.

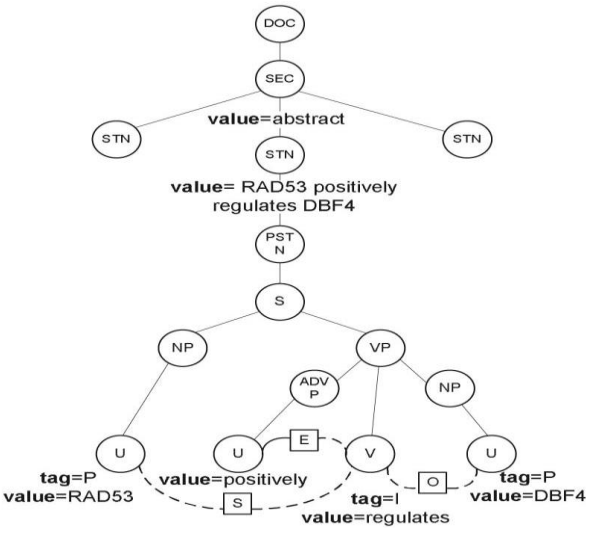

Fig5.4: An example of a parse tree for a document, which includes sections of the document

The linkage contains three different links: the S link connects the subject-noun RAD53 to the transitive verb regulates, the O link connects the transitive verb regulates to the direct object DBF4 and the E link connects the verb-modifying adverb positively to the verb regulates. The square box on a dotted line indicates the link type between two words. Each leaf node in a parse tree has value and tag attributes. The value attribute stores the text representation of a node, while the tag attribute indicates the entity type of a leaf node. For instance, a protein is marked with a tag P, a drug name with a tag D, and an interaction word is marked with I.

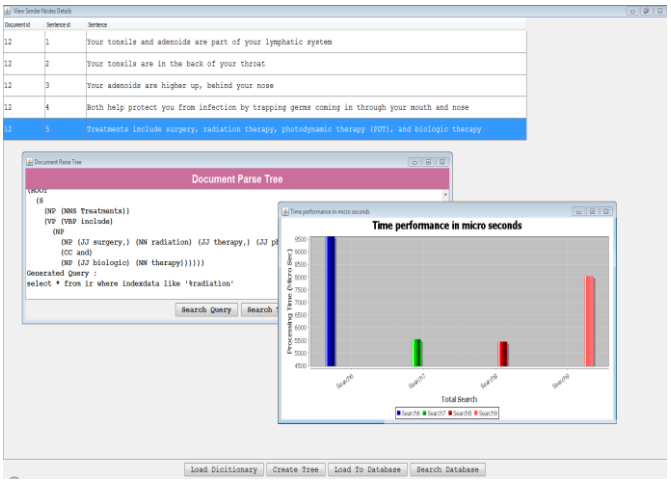

Fig 8.11: Snapshot of Time performance of search query without PTQL

3. CONCLUSIONS

Information extraction has been an active research area over the years. The main focus has been on improving the accuracy of the extraction systems, and IE has been seen as a one-time execution process. Such paradigm is inadequate for real-world applications when IE is seen as long running processes. We describe how our proposed extraction framework differs from traditional IE systems, rule-based IE systems and IE systems based on machine learning. While new documents can be added to our text collection, the content of the existing documents are assumed not to be changed, which is the case for Medline abstracts. Our focus is on managing the processed data so that in the event of the deployment of an improved component or a new extraction goal, the affected subset of the text corpus can be easily identified.

The filtering process utilizes the efficiency of IR engines so that a complete scan of the parse tree database is not needed without sacrificing any sentences that should have been used for extraction. Furthermore, our approach provides automated query generation components so that casual users do not have to learn the query language in order to perform extraction. To demonstrate the feasibility of our

incremental extraction approach, we performed experiments to highlight two important aspects of an information extraction system: efficiency and quality of extraction results.

REFERENCES

[1] Luis tari, Phan Huy Tu, Jorg Hakenberg, Yi Chen, Tran Cao Son,Graciela Gonzalez, and Chitta Baral "Incremental information extraction using Relational Databases," Vol 24 No. 1,Jan 2012.

[2] Agichtein E. and Gravano L."Querying Text Databases for Efficient Information Extraction,"Proc. Int'l Conf. Data Eng. (ICDE). 2003

[3] Banko M., Cafarella S.M., Soderland M., Broadhead S. and Etzioni O. "Open Information Extraction from the Web," Proc Joint Conf. Artificial Intelligence.

[4] Sarawagi S. "Information Extraction," Foundations and Trends in Databases, vol. 1, no. 3. 2008.