Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 11, November 2015, pg.295 – 300

RESEARCH ARTICLE

Efficient Information Retrieval System using Incremental Approach

Dr. Bode Prasad[1], D. Raveendra[2], P. Prudhvi Kiran[3]

[1]Associate Professor, IT Department, VIIT, Visakhapatnam, India
[2]PG Scholar, IT Department, VIIT, Visakhapatnam, India
[3]Assistant Professor, IT Department, VIIT, Visakhapatnam, India
[1] prasad_bode@yahoo.com; [2] raveendra4002@gmail.com; [3] pasamprudhvi@gmail.com

Abstract*— Information Retrieval Systems [12][19] are traditionally implemented as a pipeline of special-purpose processing modules targeting the extraction of a particular kind of information. A major drawback of such an approach is that 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 only a small part of the corpus might be affected. In this paper, we describe a novel approach for information extraction in which extraction needs are expressed in the form of database queries, which are evaluated and optimized by database systems. Using database queries for information extraction enables generic extraction and minimizes reprocessing of data by performing incremental extraction to identify which part of the data is affected by the change of components or goals. 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 [11] of our incremental extraction [18] approach, we performed experiments to highlight two important aspects of an information extraction system: efficiency and quality of extraction[5] results. By applying our methods to a corpus of 17 million biomedical abstracts, our experiments show that the query performance is efficient for real-time applications. Our experiments also revealed that our approach achieves high quality extraction results.

***Keywords*— Information Retrieval Systems, PTQL [1][4], PTDB, Parse tree[6], Dictionary**

I. INTRODUCTION

Information extraction (IE) is the task of automatically extracting structured information[2] from unstructured and/or semi-structured machine-readable documents. Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as information extraction.

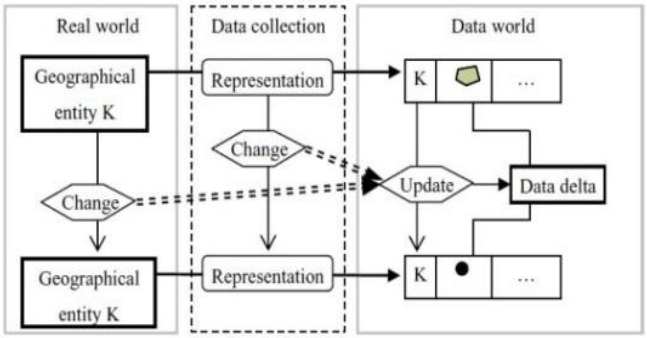

Fig. 1.1 The relationship among change and increment

Firstly, we explain the concept "geographical feature". The concept commonly is called as geographical entity in the real world and as geographical object in the world. The data delta resulted from every update operation is exclusive and certain. However, different change types correspond to different update operations. So the different change produces different data delta.

Reasons for Increment

There are three kinds of subjective changes, the changes coming from the errors amendment, the changes coming from the change of correction rules and the change of database scheme. Subjective changes of spatial data are rule less and have no uniform expression.

Objective Changes

There are many kinds of objective change in the real world. In the real world, different geographical feature with the same event may generate different changes. So, we give a new change expression based on the event type and the data delta. In this work, we only consider the simple events, that is to say, the event which affects an attribute of the geographical feature. But in the real world, there are many complex events. We can compound the generated events based on the changes of the simple events.

II. LITERATURE REVIEW

Vrushali Patil,et al**[1]**, wrote "Survey on Incremental Information Extraction Using Relational Database" it defines Information Extraction is the task of automatically extracting structured information from unstructured or semi-structured machine readable documents. In which extraction are expressed in the form of database query and query evaluated by using database system. Information extraction provides automated query generation components so that casual user does not have to learn query language in order to perform extraction. In Information extraction goal must be in the form of database query. Query must be evaluated and optimized by database system. Information Extraction is an active research area that uncovers information from large collection of text. We performed experiments to highlight two important aspect of an information extraction system: Efficiency and Quality of extraction result. Our experiment show that query performance is efficient for real time application. Our approach is composed of two phases first initial phase for processing of text and second Extraction phase for using database query to perform extraction. Incremental Extraction approach reduces processing time 90 percent as compare to traditional approach.

Luis Tari, et al**[2]**, wrote "GenerIE: Information Extraction Using Database Queries". It defines Information extraction systems are traditionally implemented as a pipeline of special-purpose processing modules. A major drawback of such an approach is that whenever a new extraction goal emerges or a module is improved, extraction has to be re applied from scratch to the entire text corpus even though only a small part of the corpus might be affected. In this demonstration proposal, we describe a novel paradigm for information extraction: we store the parse trees output by text processing in a database, and then express extraction needs using queries, which can be evaluated and optimized by databases. Compared with the existing approaches, database queries for information extraction enable generic extraction and minimize reprocessing. However, such an approach also poses a lot of technical challenges, such as language design, optimization and automatic query generation.

We will present the opportunities and challenges that we met when building GenerIE, a system that implements this paradigm.

Rajula Srilatha, et al**[3]**, wrote "A Novel Incremental Information Extraction Using Parse Tree Query Language and Parse Tree Databases". it defines Mining is nothing but retrieving the information from various resources .We have different approaches to retrieve these information one of them is traditional pipeline approach. As of increasing technologies it became more complicated[14] to workout with these traditional approach the main drawback in these pipeline approach is if any modifications are done or any module is developed newly then we have to reapply the extraction .So we are developing the different approach for data mining in this paper is through database queries . These are optimized by databases that make this as efficient approach.

After surveying these papers the major problem we have identified is, user is facing more overhead when following this traditional approaches. We present a novel approach that leads to the effective methodologies in information retrieval hierarchy, which leads to less overhead.

III.DESIGN & IMPLEMENTATION

In this paper we present a robust parsing algorithm based on the link grammar formalism for parsing natural languages. Our algorithm is a natural extension of the original dynamic programming recognition algorithm which recursively counts the number of linkages between two words in the input sentence. The modified algorithm [5] uses the notion of a null link in order to allow a connection between any pair of adjacent words, regardless of their dictionary definitions. The algorithm proceeds by making three dynamic programming passes. In the first pass, the input is parsed using the original algorithm which enforces the constraints on links to ensure grammaticality. In the second pass, the total cost of each substring[16] of words is computed, where cost is determined by the number of null links necessary to parse the substring. The final pass counts the total number of parses with minimal cost. All of the original pruning techniques have natural counterparts in the robust algorithm. When used together with memorization, these techniques enable the algorithm to run efficiently with cubic worst-case complexity.

Flowchart:

Following flowchart explains the detailed implementation of proposed work.

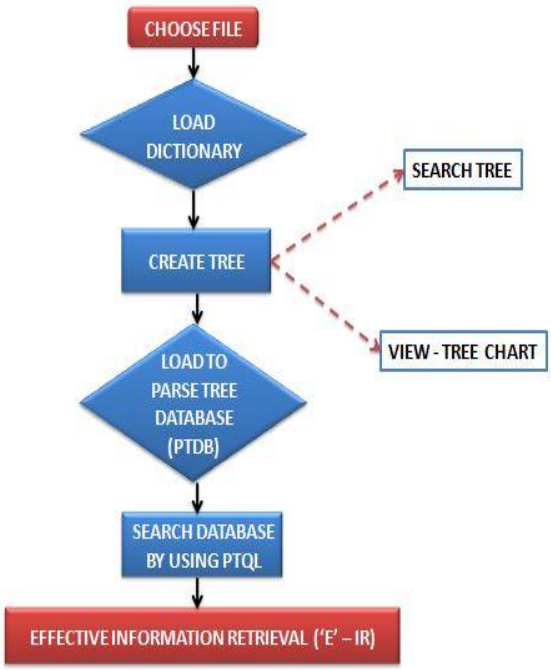

Fig. 3.1 explains the detailed implementation of proposed work

Initially we choose the raw data set, here we choose from bio medical data set (General Patient/ Diseases/ Drugs). Selected data module will be cleansed and redressed using dictionary. Various dictionary objectives include: splitting, context free grammar clustering... Etc. The redressed data is aligned in to tree data structure.

We proposed two operations that can be done on this data structure, **Search Tree** and **View – Tree chart.** Search Tree facilitates user to search the redressed data[13], which leads to more relevant information retrieval than the search done on initial raw data. View- Tree chart shows the user's overhead in a graphical and precise manner (**x axis** – Processing Time **y axis –** No. of searches).

As a next step, this data is stored in to parse tree database (**PTDB**)[1]. The main objective of this step, which is also the essence of this paper is, in the word of this information, update is a frequent operation. When the search is performed, the updated data (updated lively, during the search process) should be obtained by user. By using PTQL[1][2] and PTDB hierarchy, the processing time can be reduced. Processing time in the sense, when the data is updated, there is no need to process whole information (whole in the sense both updated and existing information). Just the updated information will be processed. But in the existing traditional methods the search will be taken place on existing data also, when the new data is updated. This leads to more user overhead.

IV. RESULTS

We first illustrate the performance of our approach in terms of query evaluation and the time savings achieved through incremental extraction. Then we evaluate the extraction performance for our two approaches in query generation.

TABLE 4.1
TIME PERFORMANCE IN SECONDS FOR PTQL EVALUATION

No. of Queries	1-Durg	2-Durg	3-Durg
Q1	10	1.5	9.2
Q2	20	2.3	11.5
Q3	30	2.7	0.5
Q4	40	3.2	1.5
Q5	50	4.1	1.9
Q6	60	26.5	30.5
Q7	70	9.2	3.1
Q8	80	40.2	15.5

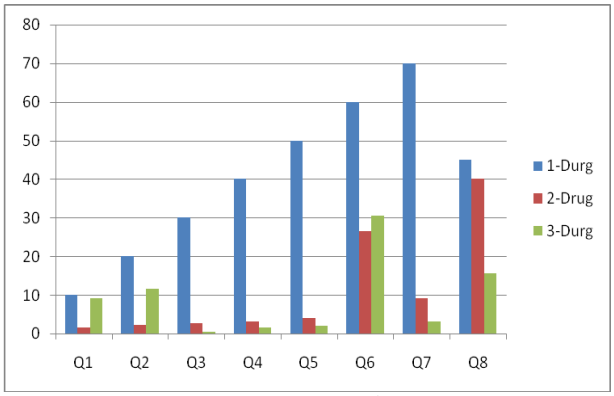

Fig. 4.1 Procedural flowchart

TABLE 4.2
COMPARISON OF DRUGS FOR PTQL EVALUATION

Number of Drugs	Normal extraction (Time in sec)	incremental Extraction (Time in sec)
0	0	0
1	0.4	0.3
3	0.6	0.5
4	0.8	0.6
5	0.9	0.7

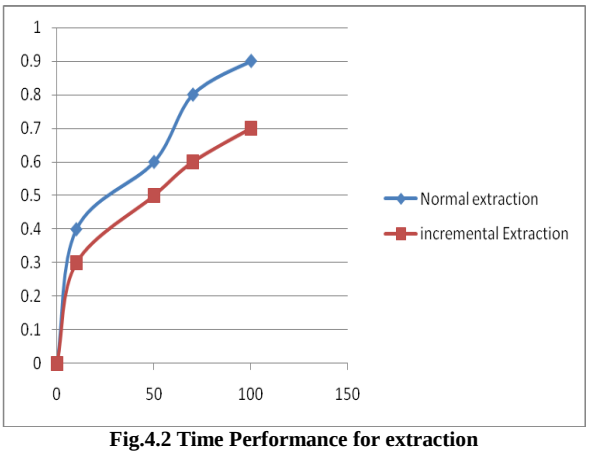

Fig.4.2 Time Performance for extraction

V. CONCLUSION

Existing extraction frameworks do not provide the capabilities of managing intermediate processed data such as parse trees and semantic information. This leads to the need of reprocessing of the entire text collection, which can be computationally expensive. On the other hand, by storing the intermediate processed data as in our novel framework, introducing new knowledge can be issued with simple SQL insert statements on top of the processed data. With the use of parse trees, our framework is most suitable for performing extraction on text corpus written in natural sentences such as the biomedical literature. As indicated in our experiments, our increment extraction approach saves much more time compared to performing extraction by first processing each sentence one-at-a time with linguistic parsers and then other components. This comes at the cost of overheads such as the storage of the parse trees and the semantic information, which takes up 1.5 TB of space for17 million abstracts for the parse tree database. In the case when the parser fails to generate parse tree for a sentence, our system generates a "replacement parse tree" that has the node STN as the root with the words in the sentence as the children of the root node. So our methodology leads to the development of effective information systems, which maintains the fewer users' overhead levels, with updated and relevant information.

REFERENCES

[1] Vrushali Patil, Prof. R.B.Wagh, "Survey on Incremental Information Extraction Using Relational Database" International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

[2] Luis Tari, Phan Huy Tu1, J¨org Hakenberg, Yi Chen, Tran Cao Son, Graciela Gonzalez, Chitta Baral, "GenerIE: Information Extraction Using Database Queries" Department of Computer Science and Engineering, Arizona State University Tempe, AZ 85287, USA

[3] Rajula Srilatha, ***K. Murali,*** **"**A Novel Incremental Information Extraction Using Parse Tree Query Language and Parse Tree Databases", International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue10 – Oct 2013

[4] F. Chen, A. Doan, J. Yang, and R. Ramakrishnan, "Efficient Information Extraction over Evolving Text Data," Proc IEEE 24th Int'l Conf. Data Eng. (ICDE '08), pp. 943-952, 2008.

[5] F. Chen, B. Gao, A. Doan, J. Yang, and R. Ramakrishnan, "Optimizing Complex Extraction Programs over Evolving Text Data," Proc 35th ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '09), pp. 321-334, 2009.

[6] S. Bird et al., "Designing and Evaluating an XPath Dialect for Linguistic Queries," Proc 22nd Int'l Conf. Data Eng. (ICDE '06), 2006.

[7] S. Sarawagi, "Information Extraction," Foundations and Trends in Databases, vol. 1, no. 3, pp. 261 377, 2008.

[8] D.D. Sleator and D. Temperley, "Parsing English with a Link Grammar," Proc Third Int'l Workshop Parsing Technologies, 1993.

[9] R. Leaman and G. Gonzalez, "BANNER: An Executable Survey of Advances in Biomedical Named Entity

[10] A.R. Aronson, "Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program," Proc. AMIA Symp., p. 17, 2001.

[11] M.J. Cafarella and O. Etzioni, "A Search Engine for Natural Language Applications," Proc. 14th Int'l Conf. World Wide Web (WWW '05), 2005.

[12] T. Cheng and K. Chang, "Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web," Proc. Conf. Innovative Data Systems Research (CIDR), 2007.
[13] H. Bast and I. Weber, "The CompleteSearch Engine: Interactive, Efficient, and Towards IR& DB Integration," Proc Conf. Innovative Data Systems Research (CIDR), 2007.
[14] S. Bird, Y. Chen, S.B. Davidson, H. Lee, and Y. Zheng, "Extending XPath to Support Linguistic Queries," Proc. Workshop Programming Language Technologies for XML (PLAN-X), 2005.
[15] J. Clark and S. DeRose, "XML Path Language (XPath)," http:// www.w3.org/TR/xpath, Nov. 1999.
[16] "XQuery 1.0: An XML Query Language," http://www.w3.org/ XML/Query, June 2001.
[17] C. Lai, "A Formal Framework for Linguistic Tree Query," Master's thesis, Dept. of Computer Science and Software Eng., Univ. of Melbourne, 2005.
[18] E. Agichtein and L. Gravano, "Querying Text Databases for Efficient Information Extraction," Proc. Int'l Conf. Data Eng. (ICDE), pp. 113-124, 2003.
[19] M. Krallinger, F. Leitner, and A. Valencia, "Assessment of the Second Biocreative PPI Task: Automatic Extraction of Protein- Protein Interactions," Proc. Second BioCreative Challenge Evaluation Workshop, 2007.

