A proposal for a Semantic Intelligent Document Repository Architecture

Alejandro Rodríguez[1], Ricardo Colomo[1], Juan Miguel Gómez[1], Giner Alor-Hernandez[2], Ruben Posada-Gomez[2], Jose Emilio Labra Gayo[3], Krishnamurthy Vidyasankar[4]

[1]Computer Science Department, Universidad Carlos III de Madrid, Spain
[2]Division of Research and Posgraduate Studies, Instituto Tecnologico de Orizaba, Mexico
[3]Computer Science Department, Universidad de Oviedo, Spain
[4]Department of Computer Science, Memorial University of Newfoundland, Canada
{alejandro.rodriguez, ricardo.colomo, juanmiguel.gomez}@uc3m.es, {galor, rposada}@itorizaba.edu.mx, labra@uniovi.es, vidya@cs.mun.ca

Abstract

The processing of high amount of documents is a highly complex challenge, which becomes even more complicated when the goal is to extract the semantically relevant data within the documents. The large-scale processing of immense repositories of knowledge requires techniques which perform information extraction to facilitate the subsequent classification and indexing of texts. Having this into account, we propose the use of Dublin Core metadata for the classification of Software Engineering publications. Based on the information obtained from Dublin Core, we present a global repository that is populated automatically, which takes the form of an ontology which represents the distinct areas of Software Engineering knowledge inspired by SWEBOK (Software Engineering Body of Knowledge). Finally, the process of the classification of texts within the ontology is carried out in three steps: keyword analysis, processing of the document. We believe our proposal based on a linguistic text classification method, heuristics, and subsequently the intersection of the three techniques mentioned, generating more precise search results in response to user queries.

1. Introduction

Multiple scientific research article repositories exist in the Internet, which is leading to disorganization on knowledge in corresponding areas. Although papers are generally classified, these classifications are very general. As a result, the domain does not represent accurately the papers topic. This paper proposes a new approach to solve these identified deficiencies, focusing on Software Engineering area. In this sense, semantic Web technologies are used in order to extract and represent explicit knowledge, relying on the standard classification described on SWEBOK. The information retrieval process is based on Dublin Core metadata, which allows document-related knowledge storage. This knowledge is represented by the use of Micro formats allowing the knowledge processing of pages with papers information. Additionally, relevance rankings are proposed that classify articles depending on the objective importance in a research area. Research papers disorganization has some consequences, such as inaccurate searches that lead to loose of time in overall processes. This approach offers also the possibility of more descriptive queries based on natural language processing and ontological queries, allowing more accurate searches over the proposed context.

The remainder of the paper is structured as follows: Section 2 presents the state of the art in similar and related technologies. Section 3 discusses the main features of the approach, the SIDRA conceptual model together with the classification algorithm and the ontology definition strategy. Finally, in Section 4 conclusions and future work are discussed.

2. Related Works

Different techniques are used to solve the considered problem. Therefore, it is convenient to explore each one separately in order to provide a broader and more comprehensible view.

There are several different solutions which attempt to solve the problems described above.

2.1 Articles classification

The classification method based on citations [1, 2] uses a backward algorithm to establish relationships between articles and create an importance measurement based on number of times articles are cited. However, these algorithms do not take into account semantic relationships between terms or the area the paper could be classified on. In his purpose, Sicilia [3], explains a description about an ontology based on SWEBOK, establishing all the classes and the relationships about the Knowledge Areas.

Learning machine-based methods [4] have become popular in document classification. In such domain, where documents are not pre-classified, some analysis and study of examples are needed in order to identify patterns between them. Support Vector Machine (SVM) is a particular technique that is suggested in [5, 6]. Specifically in [6] a study about the distribution of certain words in texts is proposed. The study evaluates the semantic distance between sentences or keywords and concepts displayed in reference ontologies. As a result, the classification is based on semantic distance between classification terms.

2.2 Information Extraction and Dublin Core Solutions

Nowadays, high amount of systems are responsible of information retrieval from the Internet. Objectives are varied and they use very different techniques to obtain this information. Increasing data structures and algorithms are appearing whose efficiency is each time greater [7, 8]. Information retrieval is a capital task in knowledge-based systems. It is applied in very different contexts, such as information retrieval from social networks [11], market studies or music classification [16]. These techniques are based on a wide range of theoretical backgrounds such as use of linear algebra or intelligent obtaining of desired information [13], which break the classic paradigm of searching lexical coincidences in texts and the queries made by the users. Other similar techniques restrict the scope in order to make comparisons with natural language [9]. Evolutive and genetic algorithms are also used to learn syntax rules and tags to filter the corresponding information [17]. Also it is important to mention the existence of architectures that make easier data filtering and retrieval tasks. As an example, some systems use these architectures to index semantic data information from the Web that provides semi structured information [12]. Besides, it is possible to find other interesting approaches focused on unstructured tests [14], which identify entities and relationships instead of identifying patterns or evaluating of presets. However, the results are not suitable compared with classical approaches. Additionally, some systems are based on language modeling [15] to extract information from Internet obtaining encouraging results. Also, another approach [26] considers the application of fuzzy logic in the information analysis that allows the relevant information identification. Furthermore, Dublin Core micro formats [18] offer a mechanism for semantic content inclusion inside Web code, in order to grant embedded metadata [19] management. The solution offers a set of tags to classify information easing up classification processes. A similar approach has been proposed in e-Learning context [10], which relies on the use of micro formats in the knowledge extraction, as well as XSL for information transformation and SPARQL for query management.

2.3 Ranking

Popularity of Web items is established by different means; independently if the item is a document, a Web page, a multimedia resource, etc. As specified before, in the research context, the popularity is calculated based on citations [20, 21].

However, a relevance evaluation could be based on ranking [22] the hosting Web posses. A Web page importance depends on both, the number of sites that links it and the relevance the linking sites have. The Figure 1 shows the corresponding algorithm:

$$PR(A) = (1-d) + d * (\frac{PR(T1)}{C(T1)} + ... + \frac{PR(Tn)}{C(Tn)})$$

Figure 1. Page Rank Formula

PR(A) is the resulting Web page *A* PageRank; *PR(Ti)* is the *i* PageRank, being *i* the Web pages that link to *A*; *C(Ti)* is the number of out links on *i* page; *d* is a stabilizing factor between 0 and 1.

3. SIDRA Solution

The solution here proposed for the stated problem reflects a hybrid system that encompasses the various resources mentioned in the previous point. In next sections, the overall system architecture and its internals are presented.

3.1 System Architecture

In this section is presented the architecture of the system. It is possible to see in Figure 2 two big and different blocks that satisfy the main architecture. In every block exists some elements that should be discussed after and that provide the internal working of all the system. The architecture is based in a typical information extraction system that extracts information from a custom source and after processing and stored it in a custom file system (in this case ontology file). Both blocks (extraction and storing blocks) are connected with one way of communication, from extraction block to storing block (ontology block). The figure 2 shows the logical architecture of the system, where it can be viewed how the distinct subsystems are communicated, as well as the flow of messages which are interchanged in order to realize the storing process. In the following subsections, the internal functioning of each of these elements is described.

3.1.1 Extraction component

This component represents the main part to extract information [24] from documents on Internet. Specially, it is focused (but not limited to) on the data extraction in Web Documents such as HTML, XHTML, to mention a few [25] [26]. Other works about data extraction from Internet are proposed in [27] [28]. The extraction component is the responsible to search documents that can be interesting for extract information and parse it in order to get valid and interesting data for the system.

3.1.1.1 Parser

The parser is the component responsible of parse the documents that the extraction module obtain. It should decompose the file in valid data and it try classify this information in valid and not. This classify operation depends of the parameters introduced in the system because in some cases there are some useful information for the system that in some other case the same information is rejected. It depends both of the system configuration and the tags that system must check in order to load and retrieve the information.

3.1.1.2 RDF or OWL Descriptions

This component is the responsible of generate or create RDF or OWL tuples from the information that parser has recollected. The objective of this component is generating correct RFD or OWL tuples in order to insert it on the ontology. The content of this tuples depends of the information that parser sent to this component. This component has direct communication with Ontology driver into Ontology component in order to send the tuple.

3.1.2 Ontology component

The ontology component is the main storing component. It is represented for an ontology containing all the tuples that extraction component provides. This component has three subcomponents providing the main functionality of the system.

3.1.2.1 Ontology Driver

The ontology driver is in the top of the ontology component because is the subcomponent that establishes and makes communications with components of the architecture (concretely with the extraction component, and more in detail with the RDF/OWL subcomponent). This component receives the tuples from RDF/OWL component with the information extracted from the extraction component. This component receives the information from RDF/OWL component and it interchanges the received information with the classifier component. The classifier component is explained in the next section.

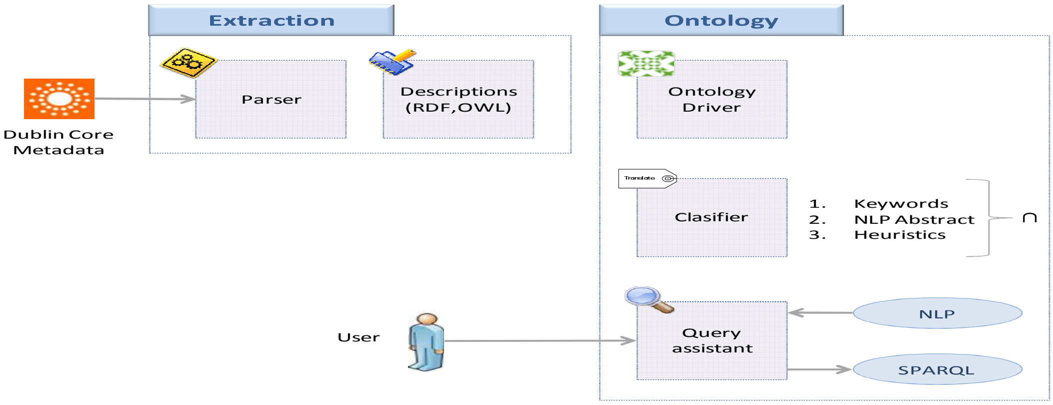


Figure 2. System architecture

3.1.2.2 Classifier

The classifier component is responsible of classify every tuple which is received by the ontology driver. Every time the ontology driver receives a new tuple, it is also sent to this component in order to classify it. To make the classification of the information contained on the tuple, the classifier uses two kind of filter:

1. **Keywords:** Classifier takes the keywords of the document in order to have more information about it.
2. **NLP Abstract:** A natural language processing (NLP) of the abstract is applied in the article in order to have more detailed information about the document.

With the intersection of the information extracted by these two filters the classifier obtains a better classification of the information contained in the tuple, obtaining better results.

3.1.2.3 Query Assistant

The query assistant is a subcomponent contained in the Ontology component, but is not directly linked with others subcomponents. Basically, this component is used to transform natural language (NL) queries through NLP techniques with the objective of create an SPARQL query.

3.2 Classification Process

The classification of documents obtained from the Web is a complex procedure, since the ambiguity of the natural language involves synonyms between two keywords, with the possibility of being in two different classes despite they reference to the same concept, the difficulty of abstracting the word true meaning caused by the word homonymy, and so on. To optimize this task, the usage of two mixed techniques has been chosen, which, as previously stated, may lead to a satisfactory outcome.

3.2.1 Keywords Analysis

The first classification mechanism is based on the analysis of the keywords referenced by each item. They indicate in a more or less reliable level, the desired covered fields, so as showing a guide to what may be the final text. Each individual of the ontology defined in their classes contains a keywords list, accepted as benchmarks to be included. The mechanism of classification is simple, as the keywords list has to be studied from the new article to be identified, so the article classification can be done. The problem with this method is twofold. Firstly, keywords in a document can refer to different kinds of classes inside the ontology, because they can be in more than one knowledge area, though it is only focused in one. On the other hand, the same word can be included in more than one class within the ontology, because it may be related to different fields of knowledge indicating whether the document is classified in one or other area with the rest of keywords, which provide clearer information on the real document topic. The problem with this classification is, more precisely, the ambiguity is not a mathematical method or a statistical models, it relies on trivial partnerships, something that does not provide a reference framework strong enough to consider this model in a unique way.

3.2.2 NLP Abstract

The Natural Language Processing (NLP) technique for the extraction of relevant content of scientific papers is defined by the progress of this technique along with Information Retrieval in the domain of the documentation. This progress is the result of research efforts in NLP and Information Retrieval (IR) which focus on synergies between both domains. NLP refers to the representation of both oral and written textual data by using theoretical models, for subsequent intelligent applications of these models within text analysis tools [29]. NLP techniques may also be used as subcomponents of systems which require a text parsing component, such as the current system. Thus, NLP applied to Science Documents data is an application of computational linguistic techniques to extract relevant elements from information textual data, for instance, the extraction of the real content of the document. In relation to previous research efforts for the classification of Science Documents, a number of test datasets for improving the techniques for text mining of this kind of documents have been provided, including in readily available XML format.

The Natural Language Processing (NLP) technique provides a more reliable result than the previous used method. This model is based on a study of the different keywords distribution within the abstract of various documents, so that for every one a graph that gives a more analytical content is generated. These graphics are compared with those texts that are already well classified within the ontology, so that in cases where there is a positive matching, it is necessary to determine that indeed, this is the class to which the document belongs (see Figure 3). This whole process is based on the GATE and JAPE architecture [30], which will draw, from a defined vocabulary, those semantically meaningful words for the domain discussed in this paper, i.e. the knowledge areas of the Software Engineering Body of Knowledge, SWEBOK.

It is also taken into account the Support Vector Machine technique, when given different arising points from the different keywords extracted from a document, it is capable of establishing the discriminating hyper plane between them, so that a more efficient and accurate classification is driven, specially compared to other more trivial methods.

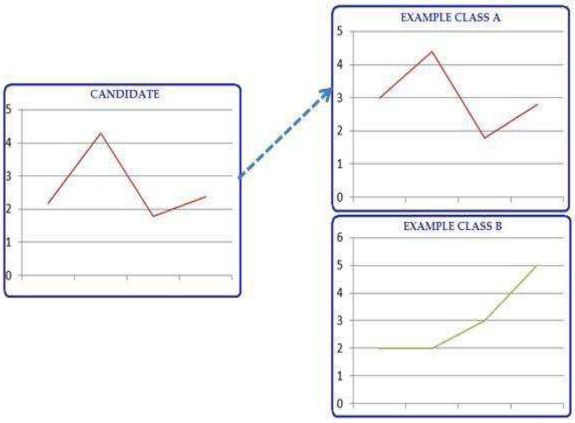


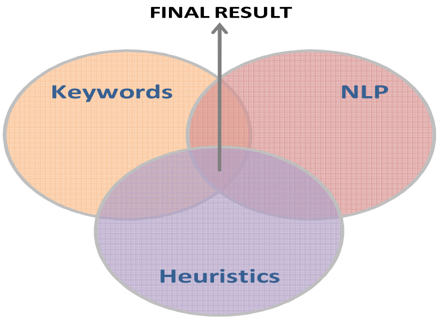
Figure 3. Example of Abstract NLP Method

3.2.3 General Classification System

The system developed states that the final outcome of the ranking is based on the intersection of the three results obtained, through the different used criteria. Therefore, the cataloging model presents a far greater consistency, with completely reliable results. With this method we managed to eliminate potential incompatibilities, weaknesses or inconsistencies of every individual model, such as the presented keywords ambiguity or the linearity of NLP. Thus, the final outcome is determined by the bounded area resulting from the three methods above mentioned, as shown in the figure 4.

3.3 Classification Ontology

The ontology is used for the various documents relating to the Software Engineering classification which is based on SWEBOK. The classes are set according to the first knowledge level of this good practice guide.



RESULT = KEYWORDS ∩ NLP

Figure 4. Classification Result

The classes covered by the ontology are as follows:

- Software Requirements
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Engineering Management
- Software Engineering Process
- Software Engineering Tools And Methods
- Software Quality

The depth explored in this system is level two, which means that it takes into account the Knowledge Areas and Subareas, without reaching very deep into topics or subtopics. For more information, it is recommended the SWEBOK query, where each of these items is explained in detail. For each one of the classes and subclasses, there is a keyword list that has to be accepted so each document is capable of being included. This eases the first classification paradigm. It has to be quoted that, a keyword does not have to be exclusively in a single class, it may belong to several knowledge areas. The list of valid keywords experiences a reverse inheritance, so we specify more it gets filtered, being much larger if it is in a class, and much more limited if it is located in an underclass. In the same way, each one of the classified items has a property that is referred to the relevance index, which has an integer type, accepting zero as the minimum value. This helps for the subsequent items management when the user performs an ontology query.

3.4 Queries

The system goal is to provide the user with an interface that allows queries performing on its repository, in order to obtain properly indexed and categorized information. The module here proposed is a wizard to carry out this process, so that the user enters a natural language query and, through NLP is transformed into a SPARQL sentence that is understandable to the ontology (see Figure 5).

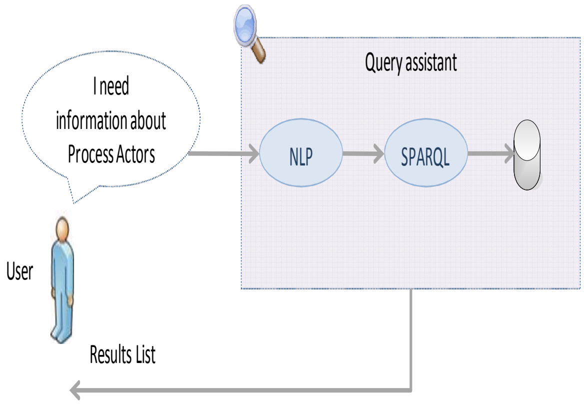


Figure 5. Query Process

For the classification, it is not an immediate process, since the query may affect a particular ontology class through its name, its keywords or the instance name. In any case to get a valid result, other sentence aspects must be analyzed. Firstly, all the stored classes / subclasses are checked, looking for matches with only the exact name, so that the result in this case would be trivial, as all the documents contained, they are returned. In the opposite case the accepted keywords for every class keywords is studied, providing the user with a list of those coincidences that respond affirmatively to the query. It is also needed to comment, due to the cardinality between keyword - class, there may be other outcomes that respond to Knowledge Area different of the user interest, which would not involve a serious problem because in every match the system shows both - the article and the related field-, so the user can make the needed document query. Anyway this should not pose a problem because the rest of the query semantics analysis could lead to a smaller grain size classification, giving more accurate results. The only real problem is that the only data to be analyzed is the search keywords entered by the user.

On the other hand, if the query process does not get positive results with this method, the search inside the articles summaries is proposed, so that any match within it is shown. This matching type would be the least reliable, because the results are less relevant than the keywords search, since a word may appear in the paper abstract but only by simple analogies, with no explicit relevance about it. In any case the processing order is the one described, so the most reliable results will be shown in the first instance, leaving those with a less reliability for a lower rank.

3.5 Article Ranking

The system uses a cataloging system in which, once an article is introduced in a given class / subclass, it also establishes an importance or relevance ratio. The items displayed to the user are based on popularity index, and as discussed in section 2, are determined by both the Page Rank and the quoted number of this article in other papers. In this way, the user have a clearer vision of each article´s importance, so as the importance the paper has taken in the scientific community, so that it can establish which documents will be more useful to the user.

4. Future Work and Conclusions

The use of powerful technologies and standard techniques give a high level of usability and can reach the objectives in an efficient and precise way in our proposal. In addition, we believe the innovations introduced like a system based on the mark obtained with the application of the three independent systems, make that the results more trustable and closer to the expected and necessary data, that represents a great improvement compared with the systems already developed.

As future work, we are interested in increase the number of systems used to know how good the information is recovered from the Web and even improve this marking system in order to extract the suitable and desired information with a more accurate percentage of good results.

5. Acknowledgements

This work is supported by the Spanish Ministry of Industry, Tourism, and Commerce under the project SONAR (TSI-340000-2007-212), GODO2 (TSI-020100-2008-564) and SONAR2 (TSI-020100-2008-665), under the PIBES project of the Spanish Committee of Education & Science (TEC2006-12365-C02-01) and the MID-CBR project of the Spanish Committee of Education & Science (TIN2006-15140-C03-02). Furthermore, this work is supported by the General Council of Superior Technological Education of Mexico (DGEST). Additionally, this work is sponsored by the National Council of Science and

Technology (CONACYT) and the Public Education Secretary (SEP) through PROMEP.

6. References

[1] Garfield, E., Citation Indexing: Its Theory and Application in Science, Technology, and Humanities, John Wiley & Sons, New York, 1979.

[2] Lawrence, S., Giles, C. L., Bollacker, K.. Digital Libraries and Autonomous Citation Indexing, IEEE Computer, Volume 32, Number 6, pp. 67-71, 1999.

[3] Sicilia, M. A., Cuadrado, J. J., García, E., Rodríguez, D., Hilera, J. R.. The Evaluation of ontological representations of the SWEBOK as a revision tool. 29th Annual International Computer Software and Applications Conference, COMPSAC, July 2005.

[4] Leouski, A. V., Croft, W. B. An Evaluation of Techniques for Clustering Search Results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, CIIR Tech. Report, 1996.

[5] Tong, S., Koller, D. Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 45-46, 2001.

[6] Deng, S., Peng, H. Document Classification Based on Support Vector Machine Using a Concept Vector Model. IEEE/WIC/ACM International Conference on Web Intelligence. Pages 473-476, 2006.

[7] Nicholas J.B. and W. Bruce Croft, Information Filtering and Information Retrieval. Two sides of the same coin?, Communications of the ACM, 1992

[8] William B. Frakes and Ricardo Baeza-Yates, Information retrieval. Data structures and algorithms, Englewood Cliffs, NJ: Prentice-Hall, 1992

[9] Johann Mitlöhner, Information Systems and e-Business Technologies, Gathering Preference Data from Restricted Natural Language, 2nd International United Information Systems Conference UNISCON 2008 Klagenfurt, Austria, April 22–25, 2008 Proceedings

[10] Ahmet Soylu, Selahattin Kuru, Fridolin Wild, Felix Mödritscher, E-Learning and Microformats: A Learning Object Harvesting Model and a Sample Application

[11] Matthew Rowe, Fabio Ciravegna, Getting to Me – Exporting Semantic Social Network Information from Facebook

[12] Andreas Harth, Jürgen Umbrich and Stefan Decker, MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data, The Semantic Web - ISWC 2006

[13] M.W. Berry, S.T. Dumais & G.W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, December 1994

[14] David C. Hooge Jr., Boanerges Aleman-Meza, I. Budak Arpinar, Entity and Relationship Extraction from Unstructured Text.

[15] Jay M. Ponte, W. Bruce Croft, A Language Modelling approach to information retrieval, Annual ACM Conference on Research and Development in Information Retrieval, 1998.

[16] Asif Ghias, Jonathan Logan, David Chamberlin, Brian C. Smith, Query by humming: musical information retrieval in an audio database, Proceedings of the third ACM international conference on Multimedia, 1995

[17] Robert M. Losee, Learning syntactic rules and tags with genetic algorithms for information retrieval and filtering: An empirical basis for grammatical rules, Information Processing & Management,Volume 32, Issue 2, March 1996, Pages 185-197

[18] Mendez, E., Lopez, LM., Siches, A., Bravo, AG. DCMF: DC & Microformats, a Good Marriage. International Conference on Dublin Core and Metadata Applications, Berlin 22-26 September 2008.

[19] Najjar, J., Wolpers, M., Duval, E. Attention Metadata: Collection and Management. Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection, Edinbug May 23, 2006.

[20] Prabowo, R., Jackson, M., Burden, P., Knoell, H. Ontology-Based Automatic Classification for the Web Pages Design, Implementation and Evaluation, Proc. Of the 3rd International Conference on Web Information Systems Engineering, 2002.

[21] Song, M., Lim, S., Kang, D., and Lee, S. Automatic Classification of Web pages based on the Concept of Domain Ontology, Proc. of the 12th Asia-Pacific Software Engineering Conference, 2005.

[22] Page, L., Brin, S., Motwani, R., Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web, Stanford Digital Libraries Working Paper, 1998.

[23] Boughanem, M., Loiseau, Y., Prade, H. Rank-Ordering Documents According to Their Relevance in Information Retrieval Using Refinements of Ordered-Weighted Aggregations. Lecture Notes in Computer Science, Volume 3877/2006. Pp 45-46. Springer 2006.

[24] Salton, MJ. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1986.

[25] Gupta, S., Kaiser, GE., Grimm, P., Chiang, MF., Starren, J. Automating Content Extraction of HTML Documents, World Wide Web, 2005, pp. 179-224.

[26] Gregg, DG., Walczak, S. Adaptive web information extraction, Communications of the ACM, 2006.

[27] Banko, M., Cafarella, MJ., Soderland, S., Broadhead, M., Etzioni, O. Open Information Extraction from the Web, Procs. of IJCAI, 2007.

[28] Etzioni, O., Cafarella, M., Downey, D., Popescu, AM., Shaked, T., Soderland, S., Weld, D. S., Yates, A. Methods for Domain-Independent Information Extraction From the Web: An Experimental Comparison, Proceedings of the national conference on artificial intelligence, 2004.

[29] Jurafsky D, Martin J H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. Prentice Hall PTR, Upper Saddle River, NJ 2008.

[30] Cunningham, H., Humphreys, K., Gaizauskas, R., Wilks, Y. Software Infrastructure for Natural Language Processing. Fifth Conference on Applied natural language processing. Washington DC, March 31-April 03, 199