International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 4 444 - 447

Survey on Software Task Extraction and Navigation

Mr. Swapnil Gangadhar Thaware
ME (IInd) year Student Computer Engineering
Vidya Pratishthan's college of engineering
Baramti Pune (India)
swapnilthaware1992@gmail.com

Prof. Dinesh Bhagwan Hanchate
Dept. of Computer Engineering
Vidya Pratishthan's college of engineering
Baramti Pune (India)
dineshbhanchate@gmail.com

***Abstract*—** In gathering requirement phase of developing software model, knowledge needed by software developer's is captured in many forms of documentation, basically it is written by different individuals. Whenever developer build a software, he struggles to find the right information in the right form at the right time so to help developers navigate documentation, In this paper we survey various techniques for extracting structured and unstructured data and automatically extracting tasks from software documentation by conceptualizing tasks as specific programming actions that have been described in the documentation. This essential information need to be extracted & annotated automatically which is challenge in software engineering.

***Index Terms*—** *Language processing, software documentation, software tasks, navigation, data extraction.*

*****

I. INTRODUCTION

Now a day's software documentation has becoming very important key factor in SDLC, (software development life cycle) while developing large as well as small scale software not only for developers but also other stakeholders like tester's, analysts, project managers, require documentation. Documentation may be in the form of softcopy of project or hardcopy and there are various types of documentation like SRS (Software requirement specification) documents, beta document, architectural and design document, technical document, marketing documents. In architecture and design documentation, there is a overview of software includes relations to an environment and construction principles to be used in design of software components. Technical documentation is having of lines of code, algorithms, interfaces, and APIs. At the end user there are manuals system administrators and support staff. In beta type of documentation there is information such as how to deliver a product into user environment and analysis of the market demand of that product and periodically taking feedback. In short software, documentation enlists enough and necessary requirement that are required for the project development, but there is still a gap between the information needs of software developers and the structure of this documentation. Any kind of structure with sections and subsections can only be enable effective navigation if the section headers are adequate clues for the information needs of developers.

II. INFORMATION EXTRACTION TECHNOLOGIES (IE)

There are mainly three different ways in which information can be extracted from a given set of documents as they are given below

1. Manual Pattern based IE :

The crux of the IE technology is certainly the power and ease of use of the way the extraction patterns are defined. Typically, the extraction patterns are defined using a formalism called regular expressions (RE). One can define a set of RE patterns for an entity. These patterns can be easily matched directly with the given input text and the matched text is extracted, which corresponds to an occurrence of that entity. For example, suppose IE task is to extract from item, the financial events that corresponds to award of contracts or projects, by one organization to another.

2. Gazette-based IE :

A rather simple approach to IE is to make use of a predefined list, that is called a gazette or gazetteer, of all possible values of a named entity. Whenever any of these examples occur in a document, they are automatically get extracted. For example, one can prepare a list of all academic degrees – Bachelor of Arts, master of along with their acronyms technology, doctor of philosophy or diploma in mechanical engineering (B.A., M.Tech., Ph.D., D.M.E.), their possible variations, for example, B.A. without the periods and abbreviations, for example, Dip. Mech. Engg.. Then it is much simpler to search the given document to find whether any of these degrees are present in it and extract them if present.

3. Machine Learning for IE :

Another powerful approach is based on the machine learning algorithms which automatically learn the IE patterns by which all occurrences of the named entity of interest are manually marked or tagged. Many supervised machine learning algorithms called classification algorithms are available which process these training examples and learn a set of rules to identify the occurrences of the named entity.

III. LITERATURE SURVEY

i) Requirements Engineering The State of the Practice [1] :

Colin J. Neill refers to prevalent, dominant and techniques used in the software development industry. In this paper, author did literature survey on web based technology. This (www.personal.psu.edu/cjn6/survey.html) website consisting of 22 questions and focused on the techniques used for requirements elicitation, representation, and modeling. There are formal and informal requirements and it can be gathered. Colin provides the main survey results in graphical format for brevity and discussed key trends and noteworthy items throughout, leaving others for the conclusions.

ii)Survey on Data Extraction of Web Pages Using Tag Tree Structure [2] :

In this paper, Vivek D. Mohod and J.V. Megha did survey on different HTML structure based technique to scrap data from web pages. We all know that internet contains large amount of data which user want to retrieve with the help of search input query. But, the result return from the web has multiple dynamic output records. Information extraction provides services to user which retrieves the information by firing query on internet.

There are two problems while extracting information mainly first: to categorize unstructured view of data. Second: to categorize structure and semi-structure view of data. Many systems need the program or the set of rules for extracting structured data from web using wrapper.

iii)Task-List Manager–A CS2 Lab on Advanced Graphical User Interface and Data Structures [3]:

In this paper Joshua Guyette and Wing H. Huen is focusing on data structure concepts to practical application tree. Tree is appropriate for easy addition and deletion of folders and task-lists. The tree contains folders (from here on called branch nodes) and task-lists (from here on called leaf nodes). The layout of the branch and leaf nodes, is an adjustable by using drag and drop method.

iv) Requirements Engineering(RE) as a Success Factor in Software Projects [4] :

Here Hubert F. Hofmann and Franz Lehner both discussed how deficient requirements can be single biggest cause of software project failure. Studying several hundred organizations, Capers Jones discovered that RE is deficient in more than 75 percent of all enterprises. In other words, getting requirements right might be the single most important and difficult part of a software project. Hubert and Franzs study provides a more integrated view of RE by investigating team knowledge, allocated resources, and deployed RE processes and their contribution to project success. In addition, Hubert incorporates the observations of previous field studies such as prioritization of requirements.

v) Information Extraction (IE) for Effective Knowledge Management [5]:

This paper highlights some of the major challenges in Information Extraction (IE) for effective knowledge management and discusses different scenarios in which the information can be extracted from a given set of documents. Information Extraction is a process of extracting information from documents containing unstructured natural language text and presenting it in a structured format. This structured information can then be effectively searched, disseminated, reused or mined using data mining techniques to discover valuable knowledge, patterns and insights. Typically, the information to be extracted from a text document consists of the following types:

(a) Generic named entities: Names of people, places, organizations, dates, times, emails, URLs, phone numbers, addresses, amounts and so on.

(b) Domain-specific named entities: Names of organisms, diseases, drugs, enzymes, proteins from biological, medical and health related documents; names of engine components in mechanical equipment maintenance and so on.

vi) TaskNav: Task-based Navigation of Software Documentation [6] : According to Christoph Treude this work is a bridge the gap between documentation structure and the information needs of software developers. For helping developers to navigate documentation, Christoph introduced TaskNav, a tool that automatically discovers and indexes task descriptions in software documentation. With TaskNav, Christoph conceptualize tasks as specific programming actions that have been described in the documentation. TaskNav presents these extracted task descriptions along with concepts, code elements and section headers in an auto-complete search interface. Output of this TaskNav is in the form of concepts, code elements and section headers.

vii) The Top Risks of Requirements Engineering[7] :

This paper tells about how worst thing that can happen in requirements engineering is gathering your set of requirements and representing customer requirement in the form of design. If

doesn't accurately represent your users needs then consequently leads your team down the wrong development path. The whole point of requirements engineering is to steer your development toward producing the right software. One of the central activities in RE is negotiating agreement on requirements. To achieve this agreement you have to find out what your customers really need. Not much of a negotiation will take place if you never actually interact with them.

viii) Natural Language Parsing of Program Element Names for Concept Extraction [8] : Here in this paper, during concept extraction to help programmers in program maintenance Brian Lawrence and Karl Wiegers present an approach which extracts concepts and relations from the source code. Brian approach applies natural language parsing to sentences constructed from the terms that appear in program element identifiers.

ix) A review of ontology based query expansion [9] : J. Bhogal and A. Macfarlane examine the meaning of context in relation to ontology based query expansion and contains a review of query expansion approaches. The various query expansion approaches include relevance feedback, corpus dependent knowledge models and corpus independent knowledge models. It analyses the use of relevance feedback, corpus dependent knowledge models and corpus independent model as ways of handling context within query expansion.

x) Query Suggestions in the Absence of Query Logs [10] : Query logs are either not available or the user base and the number of past user queries are too small to learn appropriate models. Sumit Bhatia propose a probabilistic mechanism for generating query suggestions from the corpus without using query logs. Debapriyo Majumdar utilize the document corpus to extract a set of candidate phrases. They primarily focused on the effectiveness of the queries suggested. The target systems for approach typically have smaller scale datasets and so for that purpose, the efficiency of Sumits algorithm is not critical.

IV. literature survey in Succinct

Sr. No	Author and Year	Name of paper	Paper is about	Result	Conclusion/Remark
1	Colin J. Neill and philip A. Laplante (2003)	"Requirement engineering : state of practice "	Focuses on graphical format of requirements.	Literature survey based on web and using question-nnaire and inter-view data are collected.	Survey focused on technique used for requirement and representation of requirement.
2	Vivek D. Mohod and J.V. Megha -2014	"Survey on data extraction of web pages using tag tree structure"	Data extraction technique eg.web crawler, wrapper.	Did categorization into structured and unstructured documentation.	Studied data extraction methods such as web crawler and wrapper for extracting documents.
3	Joshua Guyette and Wing H. Huen -2008	"Task-List Manager – A CS2 Lab on Advanced Graphical User Interface and Data Structures"	Applied data structure technique for task identification.	Used nested folder and task list i.e. they have created task tree.	Using hierarchical structure, things can be easily understood.
4	Hubert F. Hofmann,Franz Lehner (2001)	"Requirements Engineering as a Success Factor in Software Projects"	How deficient requirement becomes single cause of software project failure.	Collection of data is done by using questionnaires and interview from stakeholder.	Stakeholder intention are different for customer, user, project manager, analyst, developer and quality assurance.
5	TCS, White Paper by Girish Palshikar and Rajiv Sri-vastava(2012)	"Information Extraction for Effective Know-ledge Management"	Paper highlights some major challenges in information extraction.	From manual pattern based information extraction efficiently and easily done.	Total three information extraction methods such as Manual pattern based IE, Gazette based IE, Machine learning IE are used.

Sr. No	Author and Year	Name of paper	Paper is about	Result	Conclusion/Remark
6	Christoph Treude -2015	"TaskNav: Task-based Navigation of Software Documentation"	Task –Nav for navigation and preprocessing.	Output is in task, concept and code element.	Using standford NLP parser uses technique for words depen-dencies.
7	Brian Lawrence, Karl Wiegers -2001	"The Top Risks of Requirements Engineering"	Representing customer requirement in the form of design.	Different categorization of design eg. Like (LLD)Low Level Design and (HLD)High Level Design.	How worst requirement can lead project failure is explained.
8	Sumit Bhatia -2001	"Query Suggestions in the Absence of Query Logs.	Unsupervised and probabilistic approach using terms and phrases are used.	Query suggestion is to present users with queries that can lead to improv retrieval performance .	Effectiveness of the queries are suggested.

IV. CONCLUSION

We have reviewed the literature on different techniques of data extraction and navigation from web document to extract different types of information and data which is structure data and unstructured data. These techniques are based on HTML structure, some technique identifies the data record without extracting data field, and some are based on visual information to extract data. Some techniques uses DOM tree to extract repeated pattern then this repeated pattern is used to extract data.

V. ACKNOWLEDGMENTS

This paper would not have been written without the valuable advices and encouragement of Asst. Prof. D.B. Hanchate, guide of ME Dissertation work. Author's special thanks go to all the professors of computer engineering department of VPCOE Baramati, for their support and for giving an opportunity to work on survey of software task extraction and navigation.

VI. REFERENCES

[1] Colin J. Neill and Phillip A. Laplante, "Requirements Engineering :The State of the Practice," IEEE SOFTWARE Published by the IEEE Computer Society, pp. 40–45, Dec 1955.

[2] Vivek D. Mohod and Mrs. J. V. Megha, "A Survey on Data Extraction of Web Pages Using Tag Tree Structure,"IJCSIT International Journal of Computer Science and Information Technologies, pp.4361-4363, Jan 2005.

[3]] Joshua Guyette and Wing H. Huen, "Task-List Manager – A CS2 Lab on Advanced Graphical User Interface and Data Structures," 38th ASEE/IEEE Frontiers in Education Conference,pp.11-16, Oct 2008.

[4]] Hubert F. Hofmann and Franz Lehner, "Requirements Engineering as a Success Factor in Software Projects,"IEEE SOFTWARE Published by the IEEE Computer Society, pp. 58–66, July 2001.

[5] Girish K. Palshikar and Rajiv Srivastava, "Information Extraction for Effective Knowledge Management"TCS White paper, pp. 1-14.

[6] Christoph Treude and Mathieu Sicard, "TaskNav: Task-based Navigation of Software Documentation",IEEE/ACM 37th IEEE International Conference on Software Engineering, pp. 649-652, June 2015.

[7] Brian Lawrence and Karl Wiegers, "The Top Risks of Requirements Engineering,"IEEE SOFTWARE Published by the IEEE Computer Society, pp. 62–63, Dec 2001.

[8] Surafel L. Abebe and Paolo Tonella, "Natural Language Parsing of Program Element Names for Concept Extraction," 18th IEEE International Conference on Program Comprehension, pp.156-159 Feb. 2010.

[9] Bhogal J. and Macfarlane A.,"A review of ontology based query expansion", Elsevier Information Processing and Management, pp. 866-886 Oct 2006.

[10] Sumit B. and Debapriyo M, "Query Suggestions in the Absence of Query Logs," ACM SIGIR, pp. 795-804 July 2011.