International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2017, pp. 496-498
e-ISSN:2278-621X

NATURAL LANGUAGE PROCESSING IN SOFTWARE ENGINEERING

Ashwin CH[1], Akhila SP[2] & AnnapoornaShetty[3]

Abstract –The SE and NLP are fresher research territories in software engineering and building. This paper tries to raise and answer the interrelation between Software Engineering and Common Language Processing. The partners of both the examination zones which will be influenced are given. An endeavor is made to feature the likelihood of joint research in both the regions.We study the present writing, with the point of evaluating utilization of Software Engineering and Natural Dialect Processing devices in the looks into embraced. An evaluation of how different periods of SDLC can utilize NLP systems is exhibited. The paper likewise gives the avocation of the utilization of content for computerizing or joining both these ranges. A short look into course while undertaking multidisciplinary investigate is likewise given.
Keywords –Natural Language Processing (NLP), Software Engineering (SE), Software Development Life Cycle (SDLC)

1. INTRODUCTION

Software engineering and Natural language processing and connected to each other. They both are branches of computer science and engineering.Natural language processing means it is the process done many computers on natural languages.Natural language processing is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human (natural) languages, and in particular, concerned with programming computers to fruitfully process large natural language corpora.Software Engineering is a way of construction of software in a disciplined way.

It is by a firm opinion by using SE tools and technique of one research area in the context of another, better software will be developed.There is lot of many research work has been carried out in respect of Software Engineering and Natural language processing. In our research work we are trying to solve one of the question.How can SE and NLP be seen in context of each other?

2. NLP IN SOFTWARE DEVELOPMENT LIFE CYCLE

Software Development Life Cycle consists of set of phases. These phases provides guidelines to develop software. NLP can be applied in the every phases involved within the Software Development Life Cycle. It is more precise using it, when the artifacts of phase or activity are plain text. Plain text can be used as input for Natural Language Processing tasks. Basically all the activities in which the humans interpret the document there is scope of textual generation.

The below Table shows in analysis phase, which textual documents are, generated.

Analysis Phase Textual Artifacts

Document/ Artifact	**Author**
Requirement Document	System analyst
Software Requirement Specification	System analyst and Business managers
Use Case Description	System analyst
Acceptance Test Cases	Tester

The prerequisite report is created by the framework investigator in the wake of comprehension the prerequisites given by partners. Programming Requirement Specification (SRS) is a literary composed understanding between marked between the organization and the partners. Utilize cases depict the association of framework to be created with different on-screen characters [1]

3. SE IN NLP

Software Engineering involves many standard tools, processes, methodologies which can be used in the development of NLP software's.In the context of Software Engineering (SE), there are many opportunities for the application of NLP to be used to improve SE theory and practice. Recently, there have been many investigation to know the extent to which large code corpora that can be retrieved from GitHub, Stack Overflow, etc., are usable to analysis using statistical NLP models and algorithms, so

[1] Student, Department of MCA, St. Aloysius College, Mangaluru– 575022, Karnataka, India
[2] Student, Department of MCA, St. Aloysius College, Mangaluru– 575022, Karnataka, India
[3] Assistant Professor, Department of MCA, St. Aloysius College, Mangaluru– 575022, Karnataka, India

that the revolutionary advances in speech recognitions, translation, comprehension, etc. can be applied in SE. Software Development in Natural Language Processing context can be under following heading:

3.1 Open source Development-

This sort improvement constitutes giving programming with the expectation of complimentary which encourages Community improvement of programming. With regards to NLP, open source advancement liberates the engineer of programming from any legitimate edges emerging from the restrictive authorized programming. The product's multiple occasions created for look into reason and not for business improvement. The scientist's fundamental concentration is henceforth just to get the model prepared while protecting the item is left for the business.

4. ADVANTAGES OF THE INTERDISCIPLINARY RESEARCH

There are following advantages of undertaking multidisciplinary research:

a) The diverse research zones can be joined to get a more all-encompassing perspective of the basic research territory. Here for example, we are attempting to see the interdisciplinary investigate crosswise over two research territories, i.e., Software Engineering and Natural Language Handling. By tending to the issues and worries in both the territories, it is conceivable to build up a more all-encompassing methodology towards Computer Science and Engineering.
b) By attempted a joint research in both the fields, it will be conceivable to have more noteworthy plausibility of mechanization in the field of Computer Science and Engineering. This is due to mechanization it is important to have printed data or some other kind of data which is understandable to both the PC and also people.
c) By having joint research disciplines being produced, it is conceivable to accomplish all inclusive program ability.

5. COMPONENTS OF NATURAL LANGUAGE PROCESSING AND ITS MANIFESTATIONS IN PROGRAMMING ENGINEERING

There are following errands in NLP or Natural Language Understanding, which are foundational errands [2]:-

a) Morphological investigation
b) Grammatical feature (PoS)
c) Name Entity Recognition (NER)
d) Shallow and deep parsing
e) Semantics extraction
f) Pragmatics and Discourse

The above territories of NLP can be connected to Software Engineering and the other way around.

6. COMPONENTS OF SOFTWARE ENGINEERING AND ITS MANIFESTATIONS IN NATURAL DIALECT PROCESSING

The creators in the prior paper [21] have managed NLP in SE with the accompanying titles:-

a) NLP in Software Development Life Cycle
b) NLP in Umbrella Activities

The writing survey gave a knowledge into the use of Natural Language Processing to SE advancements. It was hard to attempt a Systematic Literature Review (SLR) due to different inadequacies of getting different false positives. The common dialect handling ranges can likewise be connected to different sub-zones of Software Engineering. Some of them are given beneath

a) Task Management
b) Displaying and Specification
c) Programming Testing
d) Documentation
e) Programming quality
f) Web Engineering
g) Programming Configuration Management

7. LITERATURE REVIEW

Literature Review using textual specification, domain model is generated directly. By using NLP tools such as OpenNLP and CoreNLP this work is accomplished. The overall technique involves linguistic analysis and statistical classifiers. Natural Language Text is understood by humans with little effort. The importance of textual processing on natural language text is discussed by Viliam [3]. Farid discusses the use of UML's class diagram in generation of natural language text. The paper describes various NL based systems to strengthen the view point of generating NL specification from class diagrams. The paper shows use of WordNet to clarify the structure of UML string names and generating the semantically sound sentences [5]. Reynaldo uses controlled NL text of requirements to generate class models. The paper describes some initial results arising out of parsing the text for ambiguity. The paper introduces a research plan of the author to integrate requirement validation with RAVEN project [6]. Deva Kumar, et al., created an automated tool (UMGAR) to generate UML's analysis and design models from natural language text. They have used Stanford parser, Word Net 2.1 and Java RAP to accomplish this

task [7]. Sascha, et al., proposed a round trip engineering process by creating SPIDER tool. The paper addressed the concerns about errors at requirement level being propagated to design and coding stages. The behavioral properties shown from the NL text are utilized to give developer a UML model [8]. Priya More, et al., have developed a from NL text UML Diagrams. They have developed a tool called RAPID for analyzing the requirement specifications. The software used for completing the task is OpenNLP, RAPID Stemming algorithm, WordNet [9]. Waralak, et al., discusses the role of ontology in object oriented software engineering. The author gives the introductory definition of ontology and object modeling. The paper then discusses the development tools and various standards in which ontology can be applied [10].

8. ISSUES IN UNDERTAKING JOINT RESEARCH

The issues incorporate tending to issues at Common Dialect Preparing level and Programming Building. These issues incorporate tending to the exploration level issues for both the regions. Since there are people additionally included, the human's components should likewise be mulled over.

For an entire coordination to occur, it is important to consider many issues both at the SE level and additionally at Common Dialect Preparing level. There are sure issues like human variables which can't be robotized. This being a reality, henceforth coordinating both these fields is by all accounts a removed dream. Be that as it may, look into is as yet going ahead as for robotizing the human conduct. The general accomplishment of any computerization including will rely upon the achievement of mechanizing human conduct.

9. CONCLUSION AND FUTURE SCOPE

In this paper, we attempted to build up a dream of joining SE and NLP. The writing audit being attempted is particularly in regard of creating UML graphs from Regular Language Text. The future work involves contemplating every antique of SE process models in creating more helpful data. The literary particular can be a key in giving the imperative measure of data for completing robotization. However mind should be taken to guarantee that fluctuating level of understanding does not influence the execution of the framework. Regular Language preparing and Software building albeit dissimilar, can in any case be joined with a perspective of building up a superior programming.

10. REFERENCES

[1] V. Simko, P. Kroha and P. Hnetynka, “Implemented domain model generation”, Technical Report, Department of Distributed and Dependable Systems, Report No. D3S-TR-2013-03, (2012).

[2] T. Bures, P. Hnetynka, P. Kroha and V. Simko, “Requirement Specifications Using Natural Languages, Charles University, Faculty of Mathematics and Physics”, Dept. of Distributed and Dependable Systems, Technical Report No-D3S-TR-2012-05, (2012) December.

[3] F. Meziane, N. Athanasakis and S. Ananiadou, “Generating Natural Language Specifications from UML Class diagrams”, Requirement Engineering Journal, Springer-Verlag, London, vol. 13, no. 1, (2013), pp. 1-18.

[4] R. Giganto, “Generating Class Models through Controlled Requirements”, New Zealand Computer Science Research Conference (NZCSRSC), Christchurch, New Zealand, (2008).

[5] G. Lu, P. Huang, L. He, C. Cu and X. Li, “A New Semantic Similarity Measuring Method Based on Web Search Engines”, WSEAS Transaction on Computer, ISSN: 1109-2750, vol. 9, Issue 1, (2010) January.

[6] S. Konrad and B. H. C. Cheng, “Automated Analysis of Natural Language Properties for UML Models”, [Online available], (2010).

[7] P. More and R. Phalnikar, “Generating UML Diagrams from Natural Language Specifications”, International Journal of Applied Information Systems, Foundation of Computer Science, vol. 1, no. 8, (2012)

[8] Dr. W. V. Siricharoen, “Ontologies and Object models on Object Oriented Software Engineering”, IAENG International Journal of Computer Science, IJCS, vol. 33, (2007).