Possibility of Interdisciplinary Research—Software Engineering and Natural Language Processing

Nakul Sharma, Prasanth Yalla,
KL University, nakul777@gmail.com
KL University, prasanthyalla@gmail.com

Abstract

Software Engineering and Natural Language Processing, although divergent in certain respects, do have potential in being joint research areas. In this paper, the focus is towards presenting a joint view of both these areas by utilizing the processes, tools, methods of field into another. A flowchart is presented indicating how Software Engineering and Natural Language Processing can be applied on the artifacts being in the domain of each other. The paper also provides for various issues in developing joint areas of research.

Keywords: Natural Language Understanding (NLU), Software Engineering (SE), Computational Linguistics (CL), Natural Language Processing (NLP), Software Development Life Cycle (SDLC)

1. Introduction

Software Engineering (SE) and Natural Language Processing (NLP) have both undergone many changes as result of emerging research results and their applications [1] [2]. SE and NLP both are hence constantly evolving as a result of this research. The main efforts of the researchers are to ultimately bring about the betterment of the overall field. The main intend should hence be to widen the scope of research being undertaken [35].

SE and NLP hence stand a fare chance at widening the area of research in computer science and engineering. This paper help creates a vision to combine both the research areas into a joint research. We feel that by using tools and techniques of one research area in the context of another, better software will be developed.

This work is in continuation with the author’s review paper titled “Integrating Natural Language Processing and Software Engineering”. In the previous work the effort was more directed towards seeing the two research areas into the research perspective of one another [38]. In this work, our focus is to undertake the year wise literature review keeping in view both the research areas along with the issues for undertaking joint research.

The paper is divided into following sections. Section 1 gives the brief introduction, section 2 gives the year wise literature review, section 3 gives the analysis of the existing literature, section 4 gives the issues in interdisciplinary research, section 5 provides the flowchart for doing research in an interdisciplinary environment, section 6 gives the future scope and conclusion.

2. Literature Review (Year Wise)

The literature review mainly involved searching the existing research papers of relevant information in respect to SE and NLP.

2003

Jochen L. Leidner discusses various issues in Software Engineering for natural language processing. A discussion of toolkit vs framework and system vs experiment is also given [35].
2004

Pro-case diagram from the behavioral specification is developed by Mencl et. al. The textual use cases are converted to Pro-cases based on behavioral protocols. Various case studies have been used to check the result of converting textual use cases to Pro-cases [18].

2005

Drigas et.al. have developed a system called Learning Management System (LMS) for the Greek sign language. The system provides the Greek sign language video corresponding to every text [36].

The role of use case diagrams outside the realm of software development is also discussed by Matthias et.al. The author suggests role of use case in avionics system and system engineering. The pits falls of use cases and the solutions are also presented [21].

2006

By using textual business information, UML diagrams were generated. A new methodology for extracting relevant information in natural language has been proposed and implemented. The analysis included information about the amount of objects, attributes, sequence and labeling present with respect to class, activity and sequence diagrams [14].

A speech language interface has been developed by using rule based framework. A natural language based automated tool has been used for getting the information objects and their associated attributes and methods [17].

Imran S. Bajwa et. al. presents a new model for extracting necessary information from the natural language text. The authors generate Use Case, Activity, Class and Sequence diagram from the natural language text. The designed system also allows generation of system from Natural Language Text [25].

2007

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].

Harry M Sneed has undertaken the task of developing test cases from natural language requirements. The NL text is parsed for getting the useful information such as Part-Of-Speech (POS). Using this information, test cases are generated [12].

Imran S. Bajwa et. al. propose an interactive tool to draw Use-Case diagrams. The authors have utilized LESSA approach for getting useful information from the Natural Language Text [27].

2008

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].

Arnis et. al. present a meta-model driven approach towards UML’s system as well as simulation. Authors develop the system model by identifying the artifacts from the problem domain and thereby generating Use Case and Activity diagram [22].
2010

Gang et. al. have resolved several issues in regard to word semantic similarity on web. The author make use of WordNet,’s synonym service to improve the accuracy of word similarity calculator [7].

How natural language input can be processed by a robot is shown by mapping. The paper describes language is mapped onto the structures for robot to understand [19].

Yuri et. al. have developed an Internet portal for dissemination computational linguistics knowledge and information resources. The information can be searched according to the subject content or knowledge-based navigation through the portal content [37].

2011

Imran S. Bajwa et. al. discusses an approach generating SVBR rules from Natural Language Specification. The paper shows the importance automation in generation SVBR indicating that business analyst with load of documents. They have developed an algorithm for detecting the semantics of English language [23].

Automatic generation of SVBR to UML’s class diagram is conducted with the input specification being put in SVBR format. The main issues in getting UML diagrams from SVBR are presented. Evaluation of NL tools is done using precision and recall [16].

2012

Using textual specification, domain model is generated directly by the authors. 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].

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].

Imran S. Bajwa et. al. highlights the cases in which Stanford POS tagger does not identify the particular syntactic ambiguities in English specifications of software constraints. A novel approach to overcome these syntactic ambiguities is provided and better results are presented [24].

2013

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].

Walter et.al, propose the prospect of every human to undertake programming by making universal programmability. The authors predict that by combining NLP, AI and SE, it will be possible to achieve universal programming. The authors are currently developing nlrpBENCH as a benchmark for NLP requirements [11].

Fabian Friedrich et.al. generate a process model by using natural language text. The natural text is scanned for various POS. The paper claims to make 77% of BPMN models accurately by scanning the document for necessary information [13].
BrainTool, a tool developed by Riga Technical University, has been utilized in developing UML diagrams from Natural Language Text. A manually generated UML diagrams are compared with the UML diagrams generated from the BrainTool and two-hemisphere technique [15].

2014

Mathias et.al. have developed a Requirement Feedback System (REFS) using various NLP tools and techniques. REFS generate UML Models and also checks for the feedback when the requirements are changed [34].

3. Discussion on Literature Review

There has been noticeable research in concept wise application of Software Engineering into NLP and Vice-versa. The analysis of the literature hence provided wider coverage to specific use of techniques NLP and SE. The SE has tools, methodologies, and processes etc. which are used in developing the software [29]. Table 1 gives the summary of discussion done in this paper [38].

<table>
<thead>
<tr>
<th>Paper Title</th>
<th>SE Concept/ Tools</th>
<th>NLP Tool/ Concept</th>
<th>Concept In</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>Domain Model</td>
<td>Stanford CoreNLP, Apache OpenNLP with statistical classifiers</td>
<td>SE</td>
</tr>
<tr>
<td>[34]</td>
<td>Eclipse Modelling Toolkit’s (EMF) EMFCompare</td>
<td>Autoannotator, Salmat</td>
<td>NLP and SE</td>
</tr>
<tr>
<td>[8]</td>
<td>UML Model</td>
<td>NLP Text</td>
<td>NLP and SE</td>
</tr>
</tbody>
</table>

The year wise progression in the literature provided following important observations:-
- Utilization of NLP Tools and Techniques in undertaking research on UML Diagrams
- Focus shifted more towards the developing API and automating different phases in the SDLC into a single application.
- More recent approaches had utilization of various NLP resources from which various tools have been developed.

4. Issues in Joint Research

The issues include addressing the research issues at Natural Language Processing and Software Engineering level. Since there are humans also involved, the human’s factors must also be taken into consideration [33]. Figure-1 gives the interdisciplinary issues at the SE and NLP level for generating UML diagrams from Natural Language Text.
Table 2. Issues in Joint Research for Generating UML Diagrams from Natural Language Text [39]

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Issues</th>
<th>S.E. Issues</th>
<th>NLP Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Type of UML Diagram to generated</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Generation of Textual Information</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Developing or using existing ontologies</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Human Factors</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Level of noise in a sentence</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Determining the complexity of sentence</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Scanning of Textual Information for relevant information</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Scanning of Textual Information for ambiguity</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
5. Flowchart for Interdisciplinary Research

![Flowchart](image)

**Figure 1**: Detecting Artifacts Research Areas

6. Conclusions

In this paper, an effort has been made to perceive NLP and SE as multidisciplinary areas of research. The literature review along with the issues in joint research indicates the complications associated with multidisciplinary research. It is imperative to utilize various tools to atleast bring about certain coherence at certain domains.

7. References


[31] “Shell files (sh), Linux file system’s extension. GNU Licenses.


