Survey on NLP and IR Techniques in Software Engineering

Rajat Sarin

Sayantani Goswami

Siddarth Udayakumar

1 Introduction

Our Team is working on a survey of the various NLP and IR Techniques for tasks in software engineering such as Feature/Concept Location, Bug triage, etc. The purpose of the survey is to get an inference if the conventional Information Retrieval techniques produce much better results when compared to the methods that use more modern Natural Language Processing techniques. To get our results, we got data from research papers that deal with different techniques used for different tasks, similar to the tasks mentioned above.

2 Topics

Based on the list of papers we went through, we found tasks that had a considerable amount of papers related to them and those papers also had implemented the tasks using the usual Information Retrieval techniques and Natural Language Processing techniques. The following tasks were chosen:

2.1 Concept/Feature Location

Feature or Concept location is the activity of identifying an initial location in the source code that implements a particular function in the software system.

2.2 Bug Triage

Bug triage is a process of making sure that a bug report is of high quality. This process involves making sure that the report has the necessary information for developers and makes sense.

2.3 Traceability

Software traceability is a method to discover relationships between Software artifacts to facilitate the efficient retrieval of relevant information, which is necessary for many software engineering tasks.

2.4 Source Code Pre-processing

Source code pre-processing involves extraction of identifiers in the source code, identifier separations and establishing document granularity. Once that is complete, the software system is the sent for analysis.

2.5 Automated Document Generation

Automated document generation is used in software engineering to add extra documentation notes to the software system. For example, if the generator encounters a method, a new documentation of the method is done using the verbs (if any) used in the method name.

3 Papers Used for Review

Using the spreadsheet provided to us by professor, we went through the number of papers we had for our review. The number of papers covered here include papers relating to standard information retrieval techniques and techniques that rely on Natural Language Processing.

Apart from these papers listed here, we found a few more recent papers from dealing with the topics listed down here using Google Scholar. And based on our presentation, we also added a search for papers citing other papers from the search results we obtained.

TOPIC	NUMBER OF PAPERS
Concept/Feature Localization	54
Bug Triage	22
Traceability	37
Source Code Preprocessing	13
Automatic Document Generation	26

4 Natural Language Processing techniques vs Information Retrieval techniques

Natural Language Processing (NLP) techniques for software engineering have generally been not used until recently. While going through some of the papers we have read, we noticed that NLP was used predominantly for tasks such as stop wording, stemming and other pre-processing tasks. This use of NLP results in better and efficient output. But, when using higher level methods, there is an overhead associated with processing and storage capacity. Therefore NLP has been assumed to be less ideal in some of the papers.

Information Retrieval techniques like vector space models and Latent Semantic Indexing (LSI) were found to be used more commonly. These techniques were used with variations based on the researchers' methods or they were used in combination with one or two methods or even in combination with a few

NLP methods. Our research for this survey is going to involve going through the papers, the methods used in the respective papers, compare our findings and come up with an inference based on our findings.

5 Progress

- Each of the members of the team has covered around 5 to 10 papers each. These papers deal with a particular topic (Ex: Concept Location or Bug Triage) and these tasks also deal with a particular technique (IR or NLP).
- The details of the paper which were considered for the survey include the introduction, the method or topic that was researched in the paper, the benchmarks used, results obtained and what the authors have concluded from their research.
- These details were recorded in Excel spreadsheets and Word Documents. The Excel spread sheet contains minimal details about the paper while the word documents contain more detailed information about the paper.
- The progress for all the papers covered will be updated on a spreadsheet on Github and progress will be tracked on there.

6 Milestones

We have set a few milestones for the project as follows

- Complete reading 40 more papers per team member. The papers used will be from the resources we already have and a few more recent papers will be used.
- Combine all the individual summaries obtained from our readings into one consolidated report and obtain the final inference.
- Get initial work started on the final report for the survey and report our cumulative findings on our results.

7 Schedule

TASK	PLANNED SCHEDULE
Paper reading and summarizing	~April 19
Initial work on Survey document	~April 16
Classify Results for Presentation	~ April 25
Finish Final Report	~May 5

8 Additional Comments

- During our initial presentation, we were suggested to look into more recent papers (From 2016). While we did find papers, they were few in number. We did not look for papers cited by other papers in our search. We would be including the additional papers we find in citations our research.

9 References

- Dit, Bogdan, et al. "Feature location in source code: a taxonomy and survey." *Journal of Software: Evolution and Process* 25.1 (2013): 53-95.
- Spreadsheet provided by Professor.
- Leidner, Jochen L. "Current issues in software engineering for natural language processing." *Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems-Volume 8*. Association for Computational Linguistics, 2003.
- Google Scholar (scholar.google.com)