Back to Basics: Extracting Software Requirements with a Syntactic Approach

Matthew Caron
University of Paderborn
mcaron@campus.upb.de

Frederik S. Bäumer
University of Paderborn
frederik.baeumer@upb.de

Michaela Geierhos
University of Paderborn
michaela.geierhos@upb.de

Abstract

As our world grows in complexity, companies and employees alike need, more than ever before, solutions tailored to their needs. Such tools do not always exist and need to be designed from scratch. In this paper, we present a syntactic rule-based extraction tool for software requirements specification documents. Notably, our tool allows non-expert users to express their software needs in unfiltered natural language.

1 Motivation

Since software solutions are not always available out of the box and need to be developed anew, modern techniques, such as of On-The-Fly Computing – i.e. "the automatic ad-hoc configuration of individual service compositions that fulfil customer-specific requirements" [GSB15] – provide a swifter solution to this problem and could become a genuine part of the software acquisition process. Nevertheless, projects involving custom-made solutions all share the same fundamental challenge: extracting and understanding software requirements. In truth, we regard this challenge as being twofold. First, not all users are technology experts; therefore, making the task of describing complex software requirements extremely difficult. Second, because non-expert users do not always use appropriate technical language, it can be difficult for developers to understand the genuine meaning of software requirements specification documents.

Requirements engineering, more precisely, the extraction and understanding of users' needs, plays an essential role in software development [Che12]. Requirements must be extracted, processed, analysed, and most importantly, understood by either a machine or a human agent. This challenge has been a recurring conundrum that researchers have been working hard to solve [VK08]. However, some current solutions are complicated and in some cases, even require the users to use a highly specific technical jargon when describing the desired requirements. As previously introduced, non-expert users may not have the technical expertise required to fill-out complex service templates. In this regard, we would like to present our vision of a natural language software requirements extraction approach that would allow the analysis of lengthy software requirements specification documents written in unfiltered natural language, giving non-expert users the opportunity to express themselves freely while providing developers with comprehensive, structured, and complete information.

2 Concept

With data sparsity being a major obstacle, modern approaches involving supervised or unsupervised machine learning methods are inconceivable. As a result, our architecture is based on a syntactic rule-based algorithm powered by a state-of-the-art dependency parser. In short, our approach aims at extracting every single subject-verb-object triple (SVO) from an input document. We consider this elementary approach to be an efficient and reliable way to extract actions and requirements from text and that, without the need for labelled data.

Given the extent of this research project, this study was divided into three phases. In this ongoing first phase, the focus was set, as seen in Figure 1, on retrieving SVOs, relevant adjectives, complements, and negative words. Moreover, since the order of requirements found in a document may not always represent the true logical order, we experimented with different lexicon-based techniques as a means to sort all extracted software requirements into a proper sequential order. Lastly, to address the ambiguity dilemma, it was decided that supplementary semantic information about the individual requirements should also be generated using a word sense API.

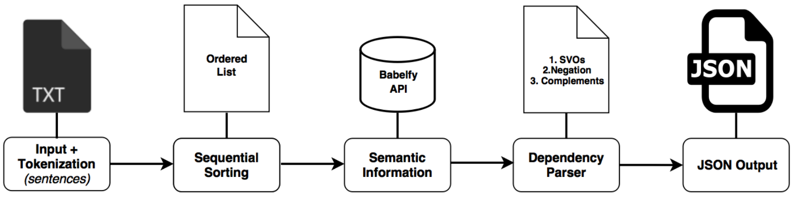

Figure 1: Architecture | Phase 1

In the second phase of this research project, the focus will be set on further refining the syntactic rule-based extraction algorithm with the help of crowdsourced data collected during the first phase. As it turns out, our tool will be made available to the public as a beta version; thus, allowing us to collect data and feedback. The acquired data will also be employed to build a classification model used for the validation of extracted software requirements. Since the SVO approach is purely syntactic, we are of the opinion that an authentication process should be added to the processing pipeline. Additionally, the possibility to replace the lexicon-based technique used to sort extracted requirements with a more contemporary solution will also be investigated.

Lastly, the third and final phase will focus on solving common problems related to the field of software requirements extraction, namely incompleteness, inconsistency, and vagueness [VK08]. As reported, requirements found in a specification document may be conflicting or even lack some valuable information. Therefore, it is clear that this matter, too, needs to be studied.

3 Conclusion

Our approach aims at facilitating software development processes by giving (non-expert) users the opportunity to voice their needs and wishes in unfiltered natural language. Because the tool's architecture is based on a syntactic rule-based approach, it has the advantage of not being domain-specific, meaning that the origin of an input document should not affect the overall performance of the extraction algorithm. As exposed, our research project will focus on overcoming common, yet arduous challenges related to software requirements engineering, namely ambiguity, incompleteness, inconsistency, and vagueness. Finally, we believe that the topic of sequential ordering is essential to any software requirements extraction task and will, therefore, be thoroughly examined.

Acknowledgements

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

References

[Che12] Murali Chemuturi. *Requirements Engineering and Management for Software Development Projects*. Springer Science & Business Media, New York, USA, 2012.

[GSB15] Michaela Geierhos, Sabine Schulze, and Frederik Simon Bäumer. What did you mean? Facing the Challenges of User-generated Software Requirements. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence*, pages 277–283, Lisbon, Portugal, 10 - 12 January 2015.

[VK08] Kunal Verma and Alex Kass. Requirements Analysis Tool: A Tool for Automatically Analyzing Software Requirements Documents. *The Semantic Web-ISWC 2008*, pages 751–763, 2008.