Towards Explainable RE Tools

Andreas Vogelsang[1]

1 Introduction

Many Requirements Engineering tasks are nowadays supported by tools that either check the quality of manual RE work or perform RE tasks completely automatic. Examples are requirements categorization [WV16], prioritization [PSA13], trace link recovery [HDO03], or detection of language weaknesses [Fe16]. The increasing abilities of these tools is driven by the availability and accessibility of complex technologies. RE tools make use of advanced natural language processing techniques [Fe16], information retrieval mechanisms [HDO03], and machine learning (e.g., by artificial neural nets [WV16]).

Despite the complex technologies used, RE tools are very appealing to practitioners because most of the technology is hidden from the user. However, when tools produce results that a user finds strange or that a user cannot explain, tools often fail to give evidence or hints *why* it made this decision and what the consequences are. Moreover, for some of the complex technologies used it may even be impossible to provide reasons for some decisions. For example, it is very hard to explain why a neural net makes a specific decision.

A special property of RE tools is that they are almost never used in a fully automated context. Most of the times, RE tools are part of processes, where they support a human analyst in performing tasks or reviewing work products. Therefore, we argue in this paper that more research is needed towards *explainable* RE tools. An explainable RE tool is able to provide rationales or indication for the decisions that it makes. Moreover, we argue that RE tools should also provide *actionable* results. A result is actionable if the tool can provide hints or recommendations on what could be done to improve or change a situation.

We use the following informal definitions of these terms: An *explainable* tool provides hints or indication on the rationale why the tool made a decision. An *actionable* tool provides hints or indication on how the user can influence the decision by changing the processed data. In our experience, most tools and approaches reported in literature are not *explainable* or not *actionable*.

[1] TU Berlin, Germanyandreas.vogelsang@tu-berlin.de

2 Towards Explainable RE Tools

In the past, we made some efforts to make our RE tools explainable and actionable. Here, we provide an example. We have developed an automated approach to differentiate requirements from non-requirements (information) in requirements documents [WV16]. At one of our industry partners, it is the document author's task to manually label all elements of a requirements document as either *requirement* or *information*. Our approach uses an artificial neural net that is trained on a large set of well-labeled requirements documents. After the training, the neural net is able to classify text fragments as one of the two classes. We use this approach to check the quality of this classification in existing documents. To make the decisions of the tool explainable, we have developed a mechanism that traces back the decision through the neural net and highlights fragments in the initial text that influenced the tool to make its decision [WV17]. As shown in Fig. 1, it appears that the word "must" is a strong indicator for a requirement, whereas the word "required" is a strong indicator for an information. While the first is not very surprising, the latter could indicate that information elements often carry rationales (why something is *required*).

Classes	requirement information
requirement	the duration until the switch is recognized as hanging must be a configurable parameter .
information	the component conditionally drives an external fan . this fan is required for active ventilation of the headlight .

Fig. 1: Explanation of the tool decision

3 Beyond Explaining: Insights through Tools

While we see increased acceptance of RE tools as the main benefit of an explainable and actionable focus, we also envision that research towards explainable RE tools may also increase our understanding on how we write and understand requirements. A good example is the use of neural networks. While it is hard to comprehend why a neural net makes specific decisions, recent research has shown that it is not impossible to analyze the inner structure of neural nets to get deeper insights into the characteristics of the learned instances. For example, Bacciu et al. [BGP16] have used so called auto encoders to force a neural net to focus on the "essential" characteristics of jokes. By training the neural net behind the auto encoder on thousands of jokes given as texts, the inner structure of the neural net had to focus on those specifics in a text that are characteristic for jokes. By analyzing the inner structure, the authors were able to identify "regions" in which the net allocates specific text fragments that it classifies as the joke's punchline or "dirty" jokes. We think that it is an interesting area of research to apply unsupervised learning techniques to large sets of requirements to learn more about the characteristics of requirements by analyzing the inner structure of the resulting networks.

References

[BGP16] Bacciu, Davide; Gervasi, Vincenzo; Prencipe, Giuseppe: LOL: An Investigation into Cybernetic Humor, or: Can Machines Laugh? In: 8th International Conference on Fun with Algorithms (FUN). 2016.

[Fe16] Femmer, H.; Méndez Fernández, D.; Wagner, S.; Eder, S.: Rapid quality assurance with Requirements Smells. JSS, 2016.

[HDO03] Hayes, J.H.; Dekhtyar, A.; Osborne, J.: Improving requirements tracing via information retrieval. In: 11th IEEE International Requirements Engineering Conference (RE). 2003.

[PSA13] Perini, Anna; Susi, Angelo; Avesani, Paolo: A Machine Learning Approach to Software Requirements Prioritization. IEEE TSE, 39(4), 2013.

[WV16] Winkler, Jonas; Vogelsang, Andreas: Automatic Classification of Requirements based on Convolutional Neural Networks. In: AIRE. 2016.

[WV17] Winkler, Jonas P.; Vogelsang, Andreas: What Does My Classifier Learn? A Visual Approach to Understanding Natural Language Text Classifiers. In: 22nd International Conference on Natural Language & Information Systems (NLDB). 2017.