Towards supporting Software Engineering using Deep Learning: A case of Software Requirements Classification

ABSTRACT

- Software Requirements are the basis of high-qualitysoftware development process, each step is related to SR, theserepresent the needs and expectations of the software in a very detailed form. The software requirement classification (SRC) taskrequires a lot of human effort, specially when there are huge of requirements, therefore, the automation of SRC have been addressed using Natural Language Processing (NLP) and Information Retrieval (IR) techniques, however, generally requires human effort to analyze and create features from corpus (set of requirements).
- In this work, we propose to use Deep Learning (DL) to classify software requirements without labor intensive feature engineering. The model that we propose is based on Convolutional Neural Network (CNN) that has been state of art in other natural language related tasks. To evaluate our proposed model, PROMISE corpus was used, contains a set of labeled requirements in functional and 11 different categories of nonfunctional requirements. We achieve promising results on SRC using CNN even without handcrafted features

EXISTING SYSTEM

- Software requirements can be classified in functional requirements and non-functional requirements.
- The functional requirements describe the procedures that the system must perform, on the other hand, non-functional requirements do not describe procedures but constrain the design of software, however, there are a few more sub-categories in non-functional requirements such as performance, maintainability, security, design.

DISADVANTAGES

- Software requirements can be classified in functional requirements and non-functional requirements. The functional requirements describe the procedures that the system must perform.
- on the other hand, non-functional requirements do not describe procedures but constrain the design of software, however, there are a few more sub-categories in non-functional requirements such as performance, maintainability, security, design,

PROPOSED SYSTEM

- we propose to use Deep Learning (DL) to classify software requirements without labor intensive feature engineering. The model that we propose is based on Convolutional Neural Network (CNN) that has been state of art in other natural language related tasks.
- To evaluate our proposed model, PROMISE corpus was used, contains a set of labeled requirements in functional and 11 different categories of nonfunctional requirements. We achieve promising results on SRC using CNN even without handcrafted features.

ADVANTAGES

- The analysis of non-functional requirements has been gained importance at first steps of software development to choose the proper architecture that satisfies them therefore, the early non-functional requirements detection plays a crucial role in software development process, mainly in the design process in order to avoid changes in further steps that may be costly.

HARDWARE REQUIREMENTS

- Processor :Intel Pentium IV 1GHz
- RAM :256MB (Min)
- Hard Drive :5GB free space
- Monitor :1024 * 768, High Color inch
- Mouse :Scroll Mouse(Logitech)
- Keyboard :104 keys

SOFTWARE REQUIREMENTS

- OS : Windows XP/7/8
- Front End : Visual Studio 2010/ netbeans 7.1
- Back End : SQL Server 2005/ heidisql 3.2
- Browser : Any Web Browser

CONCLUSION

- Based on the obtained results, it can be concluded that DL is useful on software requirements classification. The objective of using DL in this field is to avoid human effort to create and analyze features from data to improve classification, in that way the application of AI in SE could increase improving actual processes and methodologies.
- The dataset was poorly processed to became input of our model on purpose, that was to reason of measure how well DL techniques can create and transform useful features by itself to classify SR, our results show that those processes to learn from automatically generated feature representations are promising, also, it can be concluded that DL techniques to analyze text data are enough mature to be applied to real problems of another discipline without much effort.

REFERENCES

[1] Agustin Casamayor, Daniela Godoy, and MarceloCampo. "Semi-Supervised Classification of Non-Functional Requirements : An Empirical Analysis"(2009). DOI: 10.4114/ia.v13i44.1044.

[2] Joseph T Catanio. "Requirements Analysis: A Review". Advances in Systems, Computing Sciences and Software Engineering: Proceedings of SCSS05. Ed. by Tarek Sobh and Khaled Elleithy. Dordrecht: Springer Netherlands, 2006, pp. 411–418. ISBN:978-1-4020-5263-7. DOI: 10.1007/1- 4020- 5263-4 64.

[3] J Cleland-Huang et al. "The Detection and Classification of Non-Functional Requirements with Application to Early Aspects". 14th IEEE International Requirements Engineering Conference (RE'06). 2006, pp. 39–48. DOI: 10.1109/RE.2006. 65.

[4] Ishrar Hussain, Leila Kosseim, and Olga Ormandjieva. "Using Linguistic Knowledge to Classify Non-functional Requirements in SRS documents". Natural Language and Information Systems: 13thInternational Conference on Applications of Natural Language to Information Systems, NLDB 2008 London, UK, June 24-27, 2008 Proceedings. Ed. by Epaminondas Kapetanios, Vijayan Sugumaran,and Myra Spiliopoulou. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 287–298. ISBN: 978-3- 540-69858-6. DOI: 10.1007/978-3-540-69858-6 28.

[5] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". arXiv:1412.6980[cs.LG] (2014), pp. 1–15. arXiv: 1412.6980.