Automated similarity detection: identifying duplicated requirements *PEC1 - Work plan*

Quim Motger de la Encarnacin

Universitat Oberta de Catalunya
Universitat Politecnica de Catalunya

1 Introduction

This document is a detailed, exhaustive description of the scope and the work methodology plan of the master's thesis titled *"Automated similarity detection: identifying duplicated requirements"*. The following sections aim to present a general depiction of the work plan, starting from the topic proposal and refinement until the elaboration of the final thesis and its defense.

To achieve this purpose, this deliverable is structured in two chapters. First of all, the development of the thesis topic. This section provides a general description of the master's thesis, which includes the main topic, the scope of the project, its motivations, the areas of interest, and the expected results from an academical point of view. Based on this description, it also presents a detailed list of both general and specific objectives, which will be used as the general guidelines of the tasks carried out during the thesis development.

Second of all, it is necessary to address the details about the work development methodology and the schedule plan, using the previous details - such as the specific objectives - to extract concrete tasks and evaluate its development from a management perspective. Taking this plan as a reference, it is also necessary to identify possible risks that may affect the success of some of the project goals, as well as to propose mitigation techniques for these risks.

This thesis is being developed as a final master thesis inside the *Computer Engineering Master's degree* program, and therefore it is planned and developed following its quality and evaluation guidelines.

2 Thesis topic

In this section a general description of the thesis project and the goals of the project are introduced.

2.1 General description

Artificial Intelligence (AI) is a wide-known computer science area which has experienced an exponential growth both in the research field and in real use-case applicability. This computational representation of human cognitive knowledge can be represented and focused in many different areas of application, according to the features and the main goals of these areas. Two of the most known areas of AI are Machine Learning (ML), which has already been addressed during the *Advanced Artificial Intelligence* subject, and Natural Language Processing (NLP).

NLP has a large potential in different applications involving automated, computational process of all kinds of documents and textual items. This technology applies to the task of developing partial representations of features and rules of natural language based on its textual information, which includes both syntactic and semantic knowledge [**?**]. The main purpose of this technology is to use this representative knowledge in order to apply automated analysis and generation of text units, such as comprehensive sentences or full documents.

In: A. Editor, B. Coeditor (eds.): Proceedings of the NLP4RE Workshop, Essen, Germany, 18-21 March, 2019, published at http://ceur-ws.org

One of the areas of application of AI - and to be specific, NLP and ML - is the Requirements Engineering (RE) field. RE is the set of activities and processes of software engineering focused on the development, analysis, communication and management of a set of requirements that describes the features of a system [**?**]. Software development experience in recent years proves that managing and maintaining large sets of requirements have become critical issues. This problem is even more challenging due to the large amount of data and the dimensions that these projects are dealing with nowadays [**?**]. Whether the analysis and the evaluation of requirements is a tedious, time-consuming task, it is critical that they are carried out with both accuracy and efficiency in any software development project.

Between the main problems of RE we can highlight the detection and management of duplicated requirements [**?**]. If ignored, these duplicated items may lead to redundancy in the textual information of a project and therefore this may lead to the duplicity of tasks, which are critical issues from the project management perspective. Moreover, the automation of this process and the standardized usage of specific, accurate tools are still at a state-of-the-art stage. On the one hand it is difficult to find open source tools and frameworks providing generic, adaptive solutions for duplicate detection, and most of them are addressed to a very specific casuistic or use case [**?**]. On the other hand, similarity detection algorithms are highly tightened to the quality of the data used for the detection process [**?**].

This is the starting point of this master thesis: a proposal to apply automated requirement similarity detection, using artificial intelligence techniques, for the detection of duplicates between project requirements. Based on a deep research of the state-of-the-art, this master thesis must be both a portfolio and a practical evaluation of real duplicate detection scenarios in software engineering project requirements.

The details about the scope and the goals of the project are depicted in the following sections.

2.2 General objectives

Listed below are the global objectives (GO) of the project.

GO1. To research the state-of-the-art of the similarity detection field.

GO2. To develop a requirement similarity-detection multi-algorithm tool.

GO3. To prepare and to run a real application use case of duplicated requirements detection.

GO4. To evaluate and extract conclusions using the results.

2.2.1 Specific objectives

Using the list of global objectives as an input, the specific objectives (SO) are listed in the following list.

SO1. To examine the current status of similarity detection in the RE field from a general point of view.

SO2. To review and to enumerate the similarity detection tools, and to be specific the ML and NLP techniques that represent the state-of-the-art of the field.

SO3. To identify potential suitable candidates for the master thesis and the use case to be validated with.

SO4. To elaborate a development proposal for the implementation of PoC algorithms.

SO5. To develop an integrative tool to use and to test different similarity detection algorithms.

SO6. To evaluate the requirements of both input and output data of the algorithms, in order to guarantee a comprehensive analysis of the results.

SO7. To optimize and to adapt the algorithms based on the use case requirements.

SO8. To analyze and to prepare a use case dataset for all scenarios (algorithms).

SO9. To carry out a minimum, qualitatively acceptable number of experiments using the developed algorithms.

SO10. To perform a deep comparative analysis between algorithms.

SO11. To extract conclusions in terms of reliability of the results and performance of the algorithms.

3 Work plan

In order to achieve the specific objectives of this thesis, this section introduces the details of the development methodology of work, as well as the list of tasks to carry out and a schedule proposal for its achievement according to the deadlines of the project. For this purpose, a risk identification and mitigation proposal is also introduced.

3.1 Development methodology

This thesis will be developed following a Kanban-based methodology with some general influences from Scrum. This decision can be justified with three main reasons.

First of all, given the nature of the project, it seems suitable to propose an agile software development methodology to achieve the goals depicted in this document. Although this project starts from a clear, specific stage, and the objectives are detailed enough, the research of the state-of-the-art phase will deeply condition the specific tasks that will be done during the technical implementation and the evaluation process (i.e. the number and nature of algorithms). It is necessary to handle a certain flexibility in terms of requirements and task during the development of the project. Therefore, it would not be the ideal approach to use a traditional development methodology, especially after the research has been done, as requirements and tasks will need to be adapted and managed according to the advances of the work.

Second of all, this methodology aims to provide results in short-term cycles by scheduling fine-grained tasks that guarantee the success of the project's objectives. Following one of the main guidelines of Kanban, which is the visibility and traceability of the tasks, it is intended to provide a dynamic framework that will allow to complete tasks in a short period of time by identifying, detailing and scheduling them according to the general schedule planning of the master thesis (see 3.2).

Finally, and following with this last criteria of cyclic, iterative results, one of the main goals of this methodology is to hold weekly retrospective and plan meetings, comparable to analog meetings from the Scrum methodology, with the thesis director. This meetings will allow not only a constant review of the work that has been done, but an iterative review of the remaining tasks and the different potential lines of work that might raise during the development of the project.

3.1.1 Project management artifacts

- **Tasks backlog.** To Do -¿ Analyzing -¿ Doing -¿ Reviewing -¿ Done
- **Retrospective and plan meeting.**

3.2 Stage description and tasks

Following stages

Plan.-
Research.-
Development.-
Experimentation.-
Results evaluation

3.2.1 Gantt chart

3.3 Risk management

Risks

R1.
R2.
R3.

4 Conclusions

References

[1] K. Pohl, The three dimensions of requirements engineering: A framework and its applications, *Information Systems*, 19(3):243–258, 1994.

[2] O. C. Z. Gotel and C. W. Finkelstein, An analysis of the requirements traceability problem, In *Proc. of IEEE International Conference on Requirements Engineering*, pages 94–101, 1994.

[3] A. P. Nikora and G. Balcom, Automated identification of ltl patterns in natural language requirements, In *20th International Symposium on Software Re-liability Engineering*, pages 185–194, 2009.

[4] Fabiano Dalpiaz, Alessio Ferrari, Xavier Franch, Cristina Palomares: Natural Language Processing for Requirements Engineering: The Best Is Yet to Come. *IEEE Software* 35(5): 115-119 (2018)

[5] OpenReq, Project, Accessed: 2019-01-11. [Online]. Available: https://openreq.eu/.

[6] Apache OpenNLP Toolkit, Accessed: 2019-01-11. [Online]. Available:http://opennlp.apache.org

[7] NLP Toolkit for JVM Languages (NLP4J), Part-of-speech Tagging, Accessed: 2019-01-11. [Online]. Available: https://emorynlp.github.io/nlp4j/.

[8] DKPro Similarity, Accessed: 2019-01-11. [Online]. Available: https://dkpro.github.io/dkpro-similarity

[9] WordNet - A Lexical Database for English, Accessed: 2019-01-11. [Online]. Available: https://wordnet.princeton.edu/.