FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Review report of a final thesis

Reviewer:	Ing. Mgr. Ladislava Smítková Janků, Ph.D.
Student:	Bc. Oleksandr Husiev
Thesis title:	Framework for Extraction of Wikipedia Articles Content
Branch / specialization:	Web and Software Engineering, specialization Software Engineering
Created on:	March 10, 2022

Evaluation criteria

1. Fulfillment of the assignment

[1] assignment fulfilled
[2] assignment fulfilled with minor objections
▸ **[3] assignment fulfilled with major objections**
[4] assignment not fulfilled

The thesis is dominated by parts of the text inspired, quoted or directly taken as they are from other sources. The XML dump structure is not well described, the tests are described in general terms, and sometimes it is not clear from the text of the thesis how the tests were performed. In the textual parts of the thesis that present original ideas, e.g. textual parts, which are not inspired by other resources, the student often slips into overly general constructions or very vague or imprecise language.

2. Main written part

50 / 100 (E)

The basic shortcoming of the thesis is the "compilation" character of its text part. The thesis is dominated by parts of the text inspired, quoted or directly taken as they are from other sources. Even if I take into account the fact that this is a thesis in the field of software engineering, where a higher percentage of text inspired by existing technologies can be expected, there are still relatively few parts of the text that are explicitly authored. Which is a pity, because this way the thesis does not describe the student's own work in enough detail: the XML dump structure is not well described, the tests are described in general terms, and sometimes it is not clear from the text of the thesis how the tests were performed. The thesis lacks a description of the experiments performed to evaluate the usability of the thesis.

Compared to the previous version, the revised text does not show a significant change in structure, but some passages have been added and are more readable and contain more information. The mixing of the original author's text and texts taken from or inspired by other sources still persists. Compared to the previous version, the student has expanded the testing.

The student dealt with the criticism of copying more extensive texts from the non-cited sources that were part of the previous assessment in several ways: a) the text copied verbatim from the another author´ s source remained in the paper but the student added the source, b) by omitting the taken text from the paper, c) by rewriting the text in his own words, d) by adding sources cited in the text he took verbatim without citing the actual source he took.

Some of the problems indicated in the review of the previous version of the thesis remained.

For example, the text in Section 3.2 Unit Testing: in the first version of the thesis, the copied text was on page 40 ad was mentioned as "fully copied" in the review (page 40, 36 lines copied, source 011).

In this revised version of the thesis, the author does not cite the relevant wikipedia article containing a completely identical text as a source, but the sources listed as sources for the wikipedia article. Besides the fact that the takeover is not cited (he takes over from wikipedia), the question arises here whether the author actually studied the sources mentioned in the thesis or whether he just copied them from the sources of the wikipedia article. The probability that when quoting from four different sources you will compose a text completely identical to the one in wikipedia is extremely low.

Page 34: „Unit testing is a software testing method by which individual units of source code - sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures - are tested to determine whether they are fit for use."

Page 34: „The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy.
As a result, it affords several benefits[29]. Unit testing finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts
of the specification for the unit. The process of writing a thorough set of tests forces the author to think through inputs, outputs, and error conditions, and thus more crisply define the unit's desired behavior. The cost of finding a bug before coding begins or when the code is first written is considerably lower than the cost of detecting, identifying, and correcting the bug later. Bugs in released code may also cause costly problems for the end-users of the software. Code can be impossible or difficult to unit test if poorly written, thus unit testing can force developers to structure functions and objects in better ways[30][31][32]."

The first and second quoted passages are identical to passages in the wikipedia article available at https://en.wikipedia.org/wiki/Unit_testing:

"Unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage

procedures, and operating procedures—are tested to determine whether they are fit for use.[1]"

and

"The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.[1] A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.
Unit testing finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit. The process of writing a thorough set of tests forces the author to think through inputs, outputs, and error conditions, and thus more crisply define the unit's desired behavior. The cost of finding a bug before coding begins or when the code is first written is considerably lower than the cost of detecting, identifying, and correcting the bug later. Bugs in released code may also cause costly problems for the end-users of the software.[8][9][10] Code can be impossible or difficult to unit test if poorly written, thus unit testing can force developers to structure functions and objects in better ways."

The text of the thesis contains passages copied verbatim from other sources, which are not cited.
There are more such passages in the thesis, and I will give only one another example here:

page 3: "The Semantic Web is an extension of the current World Wide
Web, defined by standards set by the World Wide Web Consortium (W3C).
Although Semantic Web is a term that has often been criticized as confusing, opaque, and academic, it does nonetheless capture two of the most critical aspects of these technologies:
• Semantic: The meaning of the data is not only explicitly represented and richly expressive, but it also "travels" along with the data itself;
• Web: Individual pieces of data are linked together into a network of information, just as documents are linked together on the World Wide
Web."

The author does not cite the article containing a completely identical text as a sourcehttps://cambridgesemantics.com/blog/semantic-university/intro-semantic-web/many-names-semantic-web/

The thesis contains a small amount of original documentation for the framework created by the student. The thesis lacks preciuos documentation, e.g. precious description of the design, analysis, proposed solutions, etc. On the contrary, parts of the text inspired by or taken from other sources are predominant. In the textual parts of the thesis that present original ideas, the student often slips into overly general constructions or very vague or imprecise language.

3. Non-written part, attachments 65 / 100 (D)

Student developed a framework for extraction of Wikipedia articles content and designed tests. The student documented resources. Code is available.

4. Evaluation of results, publication outputs and awards 60 /100 (D)

I rate the framework as usable. However, the thesis lacks any description of usability testing or at least a proposal for evaluating the usability of the built framework.

The overall evaluation 55 /100 (E)

The basic shortcoming of the thesis is the "compilation" character of its text part. The thesis is dominated by parts of the text inspired, quoted or directly taken as they are from other sources. Also the text passages copied from other sources, not cited or quoted with the help of citations contained in the original sources, are present in the thesis. As far as the non-textual part is concerned, it can be stated that the student has designed and implemented the framework. I suggest a grade of E.

Questions for the defense

In the Conclusion, you mention "the article parsing success rate" as a possible improvement. Could you explain the possible technical solution in more detail?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.