Improving Systematic Literature Review Based on Text Similarity Analysis

Jingdong Jia and Xi Liu
School of Software, Beihang University, Beijing, 100083, China
E-mail: jiajingdong@buaa.edu.cn

Abstract. Systematic literature review (SLR) is an important method for identifying, evaluating, and summarizing a specific research subject. However, a SLR study always faces validity threats because there exist subjective biases in certain steps of a SLR, such as identification of relevant studies and primary study selection. In software engineering field, many systematic review studies have adopted several methods to increase the reliability, for example, group decision and pilots search. In this paper, in order to reduce subjective biases, we propose a different solution to improve objectively the quality of SLR, which aims to apply text similarity analysis into two stages of a SLR. We propose that the synonym and hypernym relations of WordNet can be used to expand the search strings to improve the paper search process. In addition, through calculating the relevance of a paper and research issues according to text similarity analysis, the primary study selection can be improved. Then, through the experiment, we validate the effectiveness of our method. Our work provides a preliminary exploration of the combination of text similarity analysis and SLR in order to improve the quality of a SLR.

1. Introduction
A research area’s mature is always accompanied by a sharp increase in the number of reports and results, and it becomes important to summarize and provide an overview [1]. Much work so far has been focused on SLR because it can help to provide such an overview. SLR is a means of identifying, evaluating and interpreting all available researches relevant to a particular research question, or topic area, or phenomenon of interest [2]. It plays a significant role in academia. For example, in software engineering field, the journal of information and software technology (IST) encourages and welcomes submissions of systematic literature studies within its scope, and the journal of IEEE transactions on software engineering lists SLR as one of the six topic areas in its aim and scope. According to our investigation, 15% (17 out of 117) of the papers published in the latest 7 volumes (93-99) in IST in 2018 are SLR papers.

Although SLR is a commonly used method in many fields, it still has limitations. The most common limitations are the possible biases introduced in the selection process and inaccuracies in the data extraction [3]. Pirzadeh pointed out that it may be affected by investigator bias during the data extraction phase in the valid threat chapter[4]. The common solution for this limitation is group decision among multiple authors; the results are cross checked and disagreements are discussed and resolved in meetings [5]. However, this solution is still a human subjective decision. In this paper, we want to improve the quality of selection and data extraction by an objective method based on text similarity analysis. Another common limitation to SLR is the difficulty in finding all relevant articles. In order to solve the question, Cruz et al. added a “snowball” search strategy in the second stage of the search process to look for relevant papers in the references [3]. This method is time-consuming, so it is
still a significant challenge. In fact, selecting suitable search strings is important for this issue, and expanding search strings can find more results. However, how to expand search strings is a question.

In order to overcome the two limitations, we want to apply text similarity analysis into SLR in this paper. It is generally accepted that text similarity analysis is a mature method in the field of natural language processing [6]. For a SLR, selection process is based on search strings, which come from the research questions. In order to obtain suitable search strings, alternative words and synonyms are used (e.g. [7-9]), but synonyms are produced subjectively in these paper. We think text similarity analysis can help solve these limitations objectively because text similarity analysis is a method of obtaining content that fits well with its own needs and interests from massive information. We have found that no related papers have combined text similarity analysis methodology with SLR yet. It is worth to explore the combination of text similarity analysis and SLR. In this paper, we attempt to conduct their combination in two stages of a SLR to improve the quality of the SLR.

The rest of this paper is structured as follows: Section 2 presents related work. Section 3 describes the research method. In Section 4, the experiment of this study is reported. Finally, the conclusions of the study are given in Section 5.

2. Related Work
In this section, we briefly summarize the results that are most relevant to our work.

2.1. Systemic Literature Review
SLR is a type of literature review that uses systematic methods to collect data, critically appraise research studies, and synthesize studies [10]. The specific steps of the SLR may differ due to the author’s writing style, but the method of SLR is roughly divided into five stages.

In the first stage of SLR, a research purpose is preset and research questions are created. Petersen [1] explained that in a SLR the existing studies related to the predefined systematic review research questions are reviewed in depth. Specific research questions are used to guide the data extraction, analysis, synthesis and presentation of results.

Selection of primary studies is the second stage. With a specific goal to retrieve conceivable significant reviews, lists of search string were adopted to match each research question [8]. Considering the comprehensiveness of selected papers, it is very necessary to investigate strings as fully as possible.

The third stage is study quality assessment. The prospective studies were required to go through inclusion and exclusion criteria [7]. In addition, the quality evaluation of the chosen studies can be accomplished by utilizing a weighting approach to examine significant studies.

The next stage is data extraction & monitoring. After completing the study selection process, a basic information will be recorded on each paper to gather information [11].

The last stage is data synthesis and analysis, which give an overall result from all of the data. This is done to answer all of the research questions.

2.2. Text Similarity Analysis
Text similarity is widely discussed in different fields. Its connotation is different due to the different application scenarios, so there is no unified and accepted definition. Lin presented a universal definition of similarity in terms of information theory derived from a set of assumptions[12]. He found that the similarity between A and B is related to their commonality and differences. The more commonality and the less differences they share, the more similar they are.

The methods of text similarity calculation are divided into four categories: string-based method, corpus-based method, knowledge-based method and other methods.

String-based method starts from the string matching degree, and measures the co-occurrence of strings and the degree of similarity. It is textual comparisons at the literal level, and textual representations are the original texts.

The corpus-based approach uses the information obtained from the corpus to calculate text similarity. Using neural network models to generate word vectors is a method for the computation of
text similarity, and the method has been studied in the field of natural language processing in recent years[13]. Many models and tools for generating word vectors have also been proposed, such as Word2Vec.

The method based on world knowledge refers to the use of a knowledge base with a normative organization system to calculate text similarity. The most commonly used ontologies are universal dictionaries, such as WordNet.

Besides the three methods mentioned above, there are several other methods for text similarity calculation, such as syntactic analysis and hybrid methods.

3. Method
Based on the characteristics of SLR and text similarity analysis, we think that they can be combined in both selection of primary studies stage and study quality assessment stage. The following two sections describe how to combine them in two stages to improve the SLR.

3.1. Improving the Search Process
Selecting primary studies is to search papers in some databases based on the search strings that are set according to the research questions. However, Kitchenham et al. confirmed that some papers had not been detected by the search through re-checked the output[14]. This situation is not uncommon, and solving this problem can effectively improve the retrieval process. Obviously, expanding the search strings can help to find more related papers.

Generally, for a SLR study, some search strings are created to retrieve papers related to research questions. Search strings are always the combination of several keywords. Text similarity analysis can be used to expand the search strings; because a search string is a phrase including several words, the similarity analysis of phrase based on words is important to expand search strings. First, we get the basic search string that comes from the research questions, and segment it into words. Second, for each notional word, its synonyms and hypernyms in WordNet will be found to obtain new keywords. Next, new search strings will be gotten by combining these new keywords. Finally, by calculating the similarity between the new strings and the basic string, suitable new search strings with high similarity are selected. This process is shown in Figure 1. During the process, how to calculate the similarity between the new and the basic string is important. The arithmetic of measuring semantic similarity between phrases based on WordNet has been put forward in literature, for example the study of [15].

Through the process, search strings are expanded. Thus, more related papers will be found.

![Figure 1. The process of expanding search string.](image)

3.2. Improving the Assessment Process of a Paper’s Quality
How to select the related studies from many search results is critical for a SLR. The existing methods for assessing the quality of papers are mainly based on the investigator’s reading and judgment on the papers. Obviously, it is subjective because the judgment depends on the investigator’s opinion. In addition, reading and judging the quality of a large number of papers can be time-consuming.

We want to adopt an objective method to evaluate the quality of the paper efficiently. Comparing the textual similarity between the research questions and the content of the paper is a feasible method. Figure 2 describes the flow chart of this method.

![Figure 2. An overview of improving the assessment process of a paper’s quality process.](image)
In order to get word vectors, Word2Vec is a suitable tool, which takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary list from the training text data and then learns vector representation of words. A paper can be used as a small corpus to calculate the semantic similarity of a string set. Then, we use the Word2Vec to convert the selected paper into a word vector set, and then select each feature word in the search string as the input to calculate the mean of the distance as text similarity. The result of this text similarity serves as a reference for the quality of the paper.

4. Experiment and Analysis
In this section, an experiment is conducted to evaluate the performance of the proposed method described in Section 3.1. The experiment was based on a systematic literature review paper of [9]. This paper examines how individual decisions in software engineering are influenced by environmental factors. Its search strings are defined as follows:

(“software engineer” OR “software developer”) AND (“environmental factor” OR “situational factor” OR “external factor” OR “contextual factor” OR “surrounding factor” OR “motivation factor” OR “de-motivation factor”).

We first choose “software engineer surrounding factor” as an example of a basic string. Then, their synonyms and hypernyms were extracted from WordNet, which is shown in Table 1.

| Basic words   | Synonyms                      | Hypernyms                        |
|---------------|-------------------------------|----------------------------------|
| software engineer | programmer, computer        | computer user, cracker, hacker, engineer |
|                | programmer, coder            |                                  |
| surrounding factor | encompassing, circumferent component, constituent, element | part, be-all_and_end-all, point |

The new set of search strings can be obtained by splicing synonyms and hypernyms in the Table 1. Then, according to the similarity arithmetic proposed in the study of [15], the strings that have high similarity to the basic string are found. For this example, the first three strings with higher similarity are: “software engineer encompassing factor”, “programmer surrounding factor” and “software engineer surrounding component”.

With the same method, the other search strings in study of [9] can also be expanded. Finally, expanded search strings are shown in Table 2. These strings are very close to the research content of the study of [9]. To verify the validity of our method, we first conducted a search by the original strings, and then searched by expanded strings in Google Scholar. The results show that some articles that did not appear in the first search appeared at a later position, and are retrieved.

| Rank | Search string                        |
|------|-------------------------------------|
| 1    | software engineer encompassing factor |
| 2    | programmer surrounding factor        |
| 3    | software engineer surrounding component |
| 4    | software engineer extraneous factor  |
| 5    | software engineer circumferent factor |
| 6    | software engineer psychological factor |
5. Conclusions
In this paper, through applying text similarity analysis, we put forward an objective method including two aspects to improve the SLR. First, in order to overcome the limitation that some related papers are not found in a SLR, we propose to expand the search strings based on synonyms and hypernyms in WordNet so as to find sufficient search results. In addition, in order to improve the assessment process of a paper’s quality, we propose to calculate the degree of similarity between the paper and the search questions. Then, we conducted an experiment to verify our method. Our results show that the application of text similarity analysis method can improve some of the steps in the SLR, thus improve the efficiency and quality of SLR.

For further work, other similarity arithmetic can be used to compare the performance of improving a SLR. Additionally, other stages of a SLR can be considered to be improved.

6. References
[1] Petersen K, Feldt R, Mujtaba S and Mattsson M 2008 Proc. 12th Int. Conf. on Evaluation & Assessment in Software Engineering (Bari) p 68
[2] Kitchenham B 2004 Procedures for Performing Systematic Reviews (Keele University Technical Report TR/SE-0401) p 1
[3] Cruz S, Silva F Q B D and Capretz L F 2015 Comput. Hum. Behav. 46 94
[4] Pirzadeh L 2010 Human Factors in Software Development: A Systematic Literature Review (M. Sc Thesis) Chalmers University of Technology
[5] Salah D, Paige R F and Cairns P 2014 Proc. 18th Int. Conf. on Evaluation & Assessment in Software Engineering (London) p 5
[6] Iyer P and Singh A 2005 Proc. 2nd India Int. Conf. on Artificial Intelligence (Pune) p 2534
[7] Khatibsyarbini M, Isa M A, Jawawi D N A and Tumeng R 2017 Inf. Softw. Technol. 93 74
[8] Beecham S, Baddoo N, Hall T, Robinson H and Sharp H 2008 Inf. Softw. Technol. 50 860
[9] Jia J, Zhang P and Capretz L F 2016 Proc. 9th Int. Workshop on Cooperative and Human Aspects of Software Engineering (Austin) p 86
[10] Ham-Baloyi W T and Jordan P 2016 Health Sa Gesondheid 21 120
[11] Jayatilleke S and Lai R 2017 Inf. Softw. Technol. 93 165
[12] Lin D 1998 Proc. 15th Int. Conf. on Machine Learning p 296
[13] Bengio Y, Ducharme R, Vincent P and Jauvin C 2003 J. Mach. Learn. Res. 3 1137
[14] Kitchenham B, Brereton O P, Budgen D, Turner M, Bailey J and Linkman S 2009 Inf. Softw. Technol. 51 7
[15] Jia J, Mo H, Capretz L F and Chen Z 2018 J. Softw.-Evol. Proc. 30 e1913