Semantic Frames for Classifying Temporal Requirements: An Exploratory Study

Aurek Chattopadhyay[a], Nan Niu[b], Zedong Peng[b] and Jianzhang Zhang[c]

[a]*National Institute of Technology Rourkela, India*
[b]*University of Cincinnati, USA*
[c]*Huangzhou Normal University, China*

Abstract

Temporal requirements express the time-related system behaviors and properties. In engineering critical systems, experience has shown that temporal requirements are the most problematic type of requirements to verify. Researchers have thus used natural language processing (NLP) techniques—most notably, part-of-speech (PoS) tagging—to develop practical classifiers to distinguish temporal requirements from non-temporal ones. In this paper, we explore frame semantics—a linguistic approach to labeling a word's role in a sentence with respect to the events of interest—to augment the temporal requirements classification task. Our experiments of 111 requirements sentences from the regulatory documents show that the best classification accuracy of 90.9% is achieved when PoS features are replaced with, rather than combined with, frame semantics features. The results suggest the promising role of semantics-augmented NLP support in an understudied requirements engineering task.

Keywords

temporal requirements classification, semantic frame parsing, regulatory requirements, NLP

1. Introduction

Natural language (NL) is the *de facto* medium for specifying requirements in industrial settings. A main advantage of NL is that it facilitates shared understanding among different stakeholders who may have different backgrounds and expertise [1, 2]. NL is also ubiquitous in the development of critical systems [3]. For example, requirements of the space mission systems developed at NASA's Jet Propulsion Laboratory (JPL) continue to be written in NL [4].

Nikora [4] shared the experience in implementing space mission software systems by highlighting that temporal requirements are the most problematic type of requirements to verify. Temporal requirements express time-related system behaviors and properties, such as safety properties asserting that nothing bad happens, liveness properties asserting that something

In: F.B. Aydemir, C. Gralha, S. Abualhaija, T. Breaux, M. Daneva, N. Ernst, A. Ferrari, X. Franch, S. Ghanavati, E. Groen, R. Guizzardi, J. Guo, A. Herrmann, J. Horkoff, P. Mennig, E. Paja, A. Perini, N. Seyff, A. Susi, A. Vogelsang (eds.): Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany, 12-04-2021

Corresponding author: J. Zhang
aurekchattopadhyay1@gmail.com (A. Chattopadhyay); nan.niu@uc.edu (N. Niu); pengzd@mail.uc.edu (Z. Peng); jianzhang.zhang2017@gmail.com (J. Zhang)
https://homepages.uc.edu/~niunn (N. Niu)
0000-0001-5566-2368 (N. Niu)

good eventually happens, and many other propositions (e.g., *P* becomes true after *Q*, *R* responds to *S* before *T*, etc. [5]).

Because accurately identifying temporal requirements can significantly reduce the effort of analyzing them for consistency, researchers have developed practical support by applying natural language processing (NLP) and machine learning (ML) techniques. Nikora [4] experimented five ML classifiers (e.g., lazy Bayesian rules) with features constructed by such NLP steps as stop word removal, stemming, and part-of-speech (PoS) tagging. The results show that PoS features positively impact classification accuracy, implying the role of NLP in building practical classifiers for separating temporal requirements from non-temporal ones.

In this paper, we explore the semantic NLP support for the temporal requirements classification task. While PoS features encode the syntactic aspects of a word in a sentence (e.g., noun, verb, adjective, etc.), we speculate that a word's semantic characteristics could provide further distinguishing power in recognizing temporal requirements. In particular, we leverage *frame semantics* [6], along with the SEMAFOR frame-semantic parser [7], to examine if and how semantic features may improve the PoS-based classification performance. Frame semantics is a theory based on how humans comprehend the roles that words take in a sentence with respect to events of interest. Requirements engineering (RE) researchers have applied this theory to different tasks [8, 9]. In our work, the focus is on integrating frame semantics into the ML classifiers of temporal requirements.

The chief contribution of our work is to extend the state-of-the-art in temporal requirements classification with semantic NLP features. Our experiments with 111 requirements sentences from the regulatory documents show that the highest classification accuracy is obtained when semantic frames completely replace PoS features. Surprisingly, the combination of PoS and frame semantics achieves lower accuracy, indicating a more effective NLP step based on frame-semantic parsing could be employed in practical settings. In what follows, we present background information in Section 2. We then detail our experimental design in Section 3, report the experimental results in Section 4, and finally, conclude the paper in Section 5.

2. Background and Related Work

2.1. Temporal Requirements Classification

The capability of distinguishing between temporal and non-temporal requirements could help reduce the effort required to trace the critical concerns from mission objectives to finer details allocated to individual software and systems components. Because temporal requirements are amenable to formal specifications [10], model checking tools such as SPIN and NuSMV can be used to verify the properties and desired behaviors expressed in temporal requirements.

In practice, only a small proportion of NL requirements are of the temporal nature. The body of space mission system requirements that was analyzed by Nikora [4] consisted of a total of nearly 7500 requirements, of which approximately 500 (6.7%) were temporal requirements. To build a practical classifier, Nikora [4] showed that word frequencies would not make good distinctions. The state-of-the-art solution that Nikora [4] reported is to preprocess (stemming, removing stop words) all requirements, and then to use both the words and the PoS tags of each sentence as features to build ML classifiers. Experimenting with the body of nearly 7500

NL requirements showed that the classifier built with lazy Bayesian rules achieved the best classification accuracy at 94.4%. Probably the most valuable operational insight is what exactly constitutes the feature space of the ML algorithms: According to Nikora's work [4], each sentence's first 200 words and those words' PoS tags shall be used as ML features. If a sentence is shorter than 200 words, then all of its words and their PoS tags shall be used.

In summary, temporal requirements classification is an important task, enabling subsequent effort of deriving LTL specifications from the NL requirements [5, 11]. As pointed out by Ryan [12], NLP's role in RE must be cautioned. We next review some RE work applying frame semantics as a NLP assistance.

2.2. Frame Semantics in Requirements Engineering

Frame semantics is a theory of linguistic meaning developed by Fillmore [6]. The basic idea is that one cannot understand the meaning of a single word without access to all the essential knowledge that relates to that word. To illustrate frame semantics, let us consider the following two sentences:

- We could use a leaky bucket algorithm to limit the bandwidth.
- The leaky bucket algorithm fails in limiting the bandwidth.

Although the sentences share certain terms (e.g., "leaky bucket algorithm" and "bandwidth") and even PoS tagging results of those terms ("noun"), the first sentence proposes a solution for a specific problem and the second sentence points out a problem. With frame-semantic parsing, the different meanings of these sentences become apparent.

- We could {use}$_{\textbf{frameName=Using}}$ {a leaky bucket algorithm}$_{\text{frameElement=Instrument}}$ {to limit the bandwidth}$_{\text{frameElement=Purpose}}$.
- {The leaky bucket algorithm}$_{\text{frameElement=Agent}}$ {fails}$_{\textbf{frameName=Success or failure}}$ {in limiting the bandwidth}$_{\text{frameElement=Goal}}$.

The **"frameName"** signifies the main event of interest, and the "frameElement" shows the argument needed to understand the event. The first sentence is about **"Using"** an "Instrument" to attain a "Purpose". The second sentence is about **"Success or failure"**, and more specifically about the **"failure"** of an "Agent" in achieving a "Goal". Such frame-semantic parsing results can be obtained automatically with state-of-the-art tools like SEMAFOR. SEMAFOR implements a statistical model [13] for determining which words in a sentence evoke what kinds of frames from FrameNet, a large collection of more than 1200 frames of the English language [14].

As shown by the previous examples, frame semantics offers the NLP support that goes beyond the syntactic level, effectively distinguishing sentences that are similar from a lexical and PoS-tagging perspective. In RE, Liaskos *et al.* [8] identified a goal model's variability concerns by relying on the set of "Agentive", "Dative", "Objective", "Factitive", "Process", "Locational", "Temporal", "Conditional", and "Extent" frames. Niu and Easterbrook [9] extracted a software product line's functional requirements from NL documents with both the PoS-tagged "verb—direct object" lexical affinities and the semantic frames characterizing the variation points. Niu *et* *al.* [10] recently developed three frame-semantic patterns to identify the "asset leakage" safety property grounded in the SysML specification.

In summary, frame semantics has been applied to support requirements elicitation and modeling tasks. Inspired by these studies, we are interested in enhancing temporal requirements classification with frame semantics.

3. Experimental Design

To answer our **research question** of: "To what extent does frame semantics enhance temporal requirements classification?", we design two enhancement mechanisms over Nikora's baseline classifier [4]. Figure 1 shows our experimental setup. For each requirements sentence, we employ Python's NLTK (https://nltk.org) to perform PoS tagging, and the SEMAFOR tool (http://www.cs.cmu.edu/~ark/SEMAFOR/) for frame-semantic parsing. Our **independent variable** is concerned with the feature representation of a given requirements sentence.

- **Baseline** uses a sentence's words and their PoS tags [4].
- **Replacement** explores text and frame semantics features, using the sentence's semantic frames in place of the PoS tags.
- **Combination** aggregates text, PoS tags, and semantic frames, grouping the textual, syntactic, and semantic attributes in the feature representation.

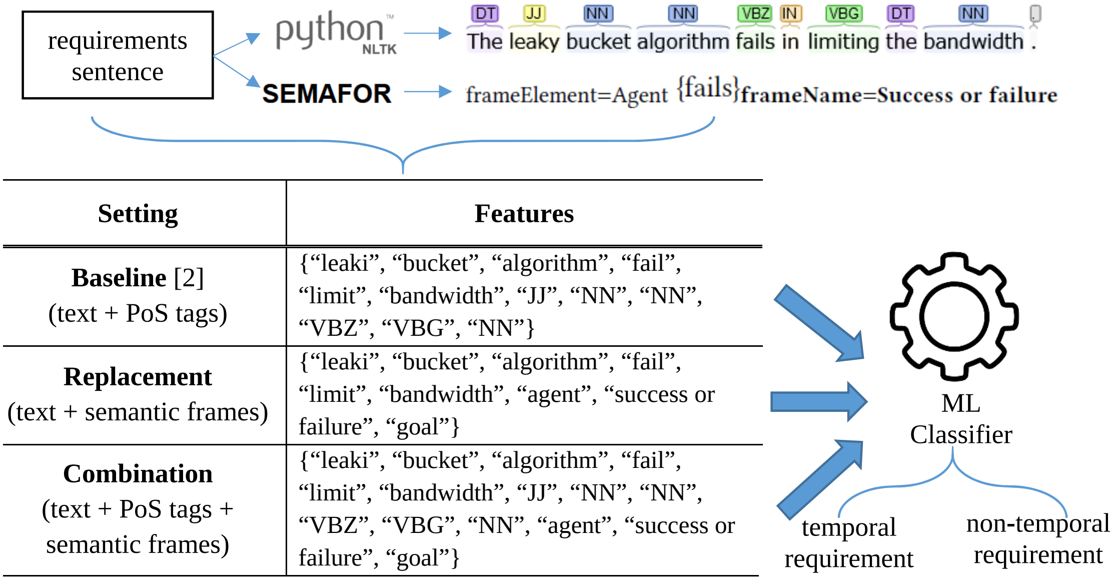
Note that our preprocessing uses Python's NLTK to remove stop words and to perform stemming. Moreover, the punctuation marks are ignored in the feature representation. Finally, following [4], if a sentence is longer than 200 words, only the first 200 words and the associated PoS and frame-semantic attributes are considered.

Once a NL requirement is represented into features, Figure 1 shows that ML classifiers are trained to classify whether the requirement is temporal or not. We have experimented four ML classifiers by using scikit-learn in Python (https://scikit-learn.org/): decision tree, logistic regression, random forest, and support vector machine (SVM).

The **dataset** that we use contains 111 NL sentences: 72 from Family Educational Rights Privacy Act (FERPA)[1] and 39 from FIPS 200[2]. FERPA is a United States federal law that governs the access to educational information and records by public entities, whereas FIPS 200 is an integral part of the risk management framework that the United States National Institute of Standards and Technology (NIST) has developed to assist federal agencies in providing levels of information security. A team of four researchers manually labeled a randomly selected 25 requirements, resulting in a substantial inter-rater agreement level with Fleiss' kappa=0.725. The discrepancies were resolved in a one-hour virtual meeting, and a labeling guideline was jointly developed. The researchers then individually labeled the remaining 86 requirements by following the guideline. The final labeling results show that, among the 111 requirements, 13 (11.7%) are temporal requirements. Table 1 lists the 13 temporal requirements.

[1] https://studentprivacy.ed.gov/sites/default/files/resource_document/file/FERPA_Enforcement_Notice_2018.pdf

[2] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.200.pdf


requirements sentence
python NLTK
SEMAFOR
DT JJ NN NN VBZ IN VBG DT NN
The leaky bucket algorithm fails in limiting the bandwidth .
frameElement=Agent {fails} frameName=Success or failure
ML Classifier
temporal requirement
non-temporal requirement

Setting	Features
Baseline [2] (text + PoS tags)	{“leaki”, “bucket”, “algorithm”, “fail”, “limit”, “bandwidth”, “JJ”, “NN”, “NN”, “VBZ”, “VBG”, “NN”}
Replacement (text + semantic frames)	{“leaki”, “bucket”, “algorithm”, “fail”, “limit”, “bandwidth”, “agent”, “success or failure”, “goal”}
Combination (text + PoS tags + semantic frames)	{“leaki”, “bucket”, “algorithm”, “fail”, “limit”, “bandwidth”, “JJ”, “NN”, “NN”, “VBZ”, “VBG”, “NN”, “agent”, “success or failure”, “goal”}

Figure 1: Each requirements sentence is preprocessed with Python’s NLTK and SEMAFOR to generate the PoS tags and the semantic frames. The sentence and the linguistic attributes are then used to represent the sentence: **Baseline** uses text and PoS tags, **Replacement** uses text and semantic frames, and **Combination** uses text, PoS tags, and semantic frames. ML classifiers are trained to identify whether the sentence is a temporal requirement or not.

We apply 10-fold cross validation for the performance evaluation. We break the dataset into 10 subsets of nearly equal size. The ML classifier is then trained with 9 subsets and tested with the remaining tenth subset. The **dependent variable** is classification accuracy, and following [4], the highest accuracy is reported. We also report the F_1-score at the class level by considering the reviewers’ comments.

4. Results and Analysis

The classification accuracy results are listed in Table 2. We note that, among the four ML classifiers, SVM and random forest have higher accuracy levels and the performances of these two are comparable. Our findings are in line with Pranckevičius and Marcinkevišius’s study [15] showing the accuracy values of SVM and random forest were similar. In addition, Alenazi *et al.* [16] showed that SVM outperformed decision tree in classifying model obstacles. While our data are NL requirements, Table 2 suggests that SVM achieves higher accuracy than logistic regression.

Compared with **Baseline**, **Replacement** achieves better performance, and such improvements are consistent across all the ML classifiers. In contrast, **Combination** performs the same as **Baseline**, indicating that the effect of semantic frames might be overshadowed by that of the PoS tags. When running the **Baseline**, the highest accuracy achieved is 86.3%, which is lower than 94.4% reported in [4]. We speculate one reason may be the smaller dataset (111 total

Table 1
Thirteen Temporal Requirements in Our Dataset

Source	Requirements Sentence
FERPA	Under FERPA, a school must provide an eligible student with an opportunity to inspect and review his or her education records within 45 days following its receipt of a request.
	A school is not required to provide an eligible student with updates on his or her progress in a course (including grade reports) or in school unless such information already exists in the form of an education record.
	If, as a result of the hearing, the school still decides not to amend the record, the eligible student has the right to insert a statement in the record setting forth his or her views.
	That statement must remain with the contested part of the eligible student's record for as long as the record is maintained.
	Under FERPA, a school may not generally disclose personally identifiable information from a minor student's education records to a third party unless the student's parent has provided written consent.
	A school may disclose personally identifiable information from education records without consent to a "school official" under this exception only if the school has first determined that the official has a "legitimate educational interest" in obtaining access to the information for the school.
	Otherwise, the school must make a reasonable attempt to notify the parent in advance of making the disclosure, unless the parent or eligible student has initiated the disclosure.
	As stated above, the conditions specified in the FERPA regulations have to be met before a school may non-consensually disclose personally identifiable information from education records in connection with any of the exceptions mentioned above.
	A timely complaint is defined as one that is submitted to the Office within 180 days of the date that the complainant knew or reasonably should have known of the alleged violation.
	Complaints that do not meet FERPA's threshold requirement for timeliness are not investigated.
	If we receive a timely complaint that contains a specific allegation of fact giving reasonable cause to believe that a school has violated FERPA, we may initiate an administrative investigation into the allegation in accordance with procedures outlined in the FERPA regulations.
FIPS 200	Organizations must establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles.
	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.

requirements) used in our experiment; however, **Replacement** obtains the best accuracy of 90.9% with three classifiers: SVM, random forest, and logistic regression. We therefore conclude that replacing the PoS features in a state-of-the-art temporal requirements classification approach [4] with semantic frames could consistently improve ML's performance.

To investigate the direct use of semantic frames to identify temporal requirements, we carried out another experiment where the candidate sentences are retrieved based on frame names (i.e., events of interest), rather than classified via supervised learning. Two researchers first manually inspected the SEMAFOR parsing results of all the temporal requirements, and then ranked the frame names according to how likely they convey temporal information in the contexts of FERPA and FIPS 200. Cumulatively, the top-ranked frame names were connected

Table 2
Classification Accuracy and Class-Level F_1 Scores ("A" is the overall accuracy, "T" is the class of temporal requirements, and "NT" is the class of non-temporal class)

	decision tree			logistic regression			random forest			SVM		
	A	T	NT	A	T	NT	A	T	NT	A	T	NT
Baseline	72.7%	46.7%	85.3%	81.8%	50.0%	88.9%	86.3%	66.7%	91.4%	86.3%	66.7%	91.4%
Replacement	81.0%	50.0%	88.0%	90.9%	80.0%	94.1%	90.9%	80.0%	94.1%	90.9%	80%	94.1%
Combination	72.7%	46.7%	85.3%	81.8%	50.0%	88.9%	86.3%	66.7%	91.4%	86.3%	66.7%	91.4%

Table 3
Cumulatively Retrieving via the Ranked Listed of Frame Names: (1) "Calendric unit", (2) "Activity ongoing", (3) "Frequency", (4) "Activity start", and (5) "Change event time"

Top-N frame names	|retrieved sentences|	|true positives|	|false positives|	|false negatives|	|true negatives|	accuracy
1	4	2	2	11	96	88.3%
2	12	3	9	10	89	82.9%
3	19	3	16	10	82	76.6%
4	21	5	16	8	82	78.4%
5	21	5	16	8	82	78.4%

by logical 'OR' to retrieve candidate sentences. The results are shown in Table 3. Although such a frame-name-based retrieval achieved the highest accuracy of 88.3%, only 2 true-positive temporal requirements were identified. In contrast, three of the trained classifiers outperformed the retrieval method, implying the synergy of textual and frame semantic features [17, 18].

A major threat affecting our exploratory study's validity is that the labeling of temporal and non-temporal requirements was done manually, though a substantial inter-rater agreement level was reached on a subset of the data. Using the requirements [19] may impact the manual labeling. We note that the NL sentences are drawn from regulatory documents, which in our opinions, are of high quality and tend to evolve on a stable basis [20]. We share all the temporal requirements used in our work in Table 1 to facilitate reuse, expansion, and replication [21, 22].

5. Concluding Remarks

Temporal requirements, though often appearing as a small fraction of NL requirements, are among the most problematic to verify. Identifying them would enable requirements engineers to formulate them into formal specifications and further leverage tools like model checking to verify them. Building on a practical ML approach [4], we have explored in this paper the support that frame semantics may offer in classifying temporal requirements. Our experiments on 111 NL requirements show that replacing syntactic features of PoS tags with semantic features results in a consistently high level of classification accuracy.

Our ongoing work tests more NL requirements from other domains (e.g., functional safety requirements from the automotive domain [23]). We are also interested in applying frame semantics to other RE activities, such as generating creative requirements [24] and requirements visualization [25]. Finally, we are investigating ways (e.g., safety patterns [10]) to derive formal specifications from the identified temporal requirements. Our goal is to offer semantic NLP support for challenging RE tasks, such as identifying and verifying temporal requirements.

Acknowledgments

We thank Shamna Abdulsalam from the University of Cincinnati for her assistance with SEMAFOR. This work was partially funded by the Research Center of E-commerce and Network Economy in Hangzhou Normal University.

References

[1] K. Pohl, Requirements Engineering: Fundamentals, Principles, and Techniques, Springer, 2010.

[2] N. Niu, S. Easterbrook, So, you think you know others' goals? A repertory grid study, IEEE Software 24 (2007) 53–61.

[3] W. Wang, A. Gupta, N. Niu, L. D. Xu, J.-R. C. Cheng, Z. Niu, Automatically tracing dependability requirements via term-based relevance feedback, IEEE Transactions on Industrial Informatic 14 (2018) 342–349.

[4] A. P. Nikora, Classifying requirements: Towards a more rigorous analysis of natural-language specifications, in: Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering, ISSRE'05, Chicago, IL, USA, 2005, pp. 291–300.

[5] A. P. Nikora, G. Balcom, Automated identification of LTL patterns in natural language requirements, in: Proceedings of the 20th IEEE International Symposium on Software Reliability Engineering, ISSRE'09, Mysuru, India, 2009, pp. 185–194.

[6] C. J. Fillmore, Frame semantics and the nature of language, Annals of the New York Academy of Sciences 280 (1976) 20–32.

[7] D. Das, N. Schneider, D. Chen, N. A. Smith, SEMAFOR 1.0: A Probabilistic Frame-Semantic Parser, Technical Report CMU-LTI-10-001, Carnegie Mellon University, 2010.

[8] S. Liaskos, A. Lapouchnian, Y. Yu, E. Yu, J. Mylopoulos, On goal-based variability acquisition and analysis, in: Proceedings of the 14th IEEE International Requirements Engineering Conference, RE'06, Minneapolis/St.Paul, MN, USA, 2006, pp. 76–85.

[9] N. Niu, S. Easterbrook, Extracting and modeling product line functional requirements, in: Proceedings of the 16th IEEE International Requirements Engineering Conference, RE'08, Barcelona, Spain, 2008, pp. 155–164.

[10] N. Niu, L. Johnson, C. Diltz, Safety patterns for SysML: What does OMG specify?, in: Proceedings of the 19th International Conference on Software and Systems Reuse, ICSR'20, Hammamet, Tunisia, 2020, pp. 19–34.

[11] S. Ghosh, D. Elenius, W. Li, P. Lincoln, N. Shankar, W. Steiner, ARSENAL: Automatic requirements specification extraction from natural language, 2016. URL: https://arxiv.org/abs/1403.3142v3.

[12] K. Ryan, The role of natural language in requirements engineering, in: Proceedings of the

1st IEEE International Symposium on Requirements Engineering, RE'93, San Diego, CA, USA, 1993, pp. 240–242.

[13] D. Das, D. Chen, A. F. T. Martins, N. Schneider, N. A. Smith, Frame-semantic parsing, Computational Linguistics 40 (2014) 9–56.

[14] C. F. Baker, C. J. Fillmore, J. B. Lowe, The Berkeley FrameNet project, in: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL'98, Montréal, Canada, 1998, pp. 86–90.

[15] T. Pranckevišius, V. Marcinkevišius, Comparison of naïve bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification, Baltic Journal of Modern Computing 5 (2017).

[16] M. Alenazi, N. Niu, W. Wang, J. Savolainen, Using obstacle analysis to support SysML-based model testing for cyber physical systems, in: Proceedings of the 8th International Model-Driven Requirements Engineering Workshop, MoDRE'18, Banff, Canada, 2018, pp. 46–55.

[17] N. Niu, X. Jin, Z. Niu, J.-R. C. Cheng, L. Li, M. Y. Kataev, A clustering-based approach to enriching code foraging environment, IEEE Transactions on Cybernetics 46 (2016) 1962–1973.

[18] W. Wang, N. Niu, M. Alenazi, J. Savolainen, Z. Niu, J.-R. C. Cheng, L. D. Xu, Complementarity in requirements tracing, IEEE Transactions on Cybernetics 50 (2020) 1395–1404.

[19] N. Niu, W. Wang, A. Gupta, Gray links in the use of requirements traceability, in: Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE'16, Seattle, WA, USA, 2016, pp. 384–395.

[20] W. Wang, F. Dumont, N. Niu, G. Horton, Detecting software security vulnerabilities via requirements dependency analysis, IEEE Transactions on Software Engineering (*accepted*). doi:10.1109/TSE.2020.3030745.

[21] N. Niu, A. Koshoffer, L. Newman, C. Khatwani, C. Samarasinghe, J. Savolainen, Advancing repeated research in requirements engineering: A theoretical replication of viewpoint merging, in: Proceedings of the 24th IEEE International Requirements Engineering Conference, RE'16, Beijing, China, 2016, pp. 186–195.

[22] C. Khatwani, X. Jin, N. Niu, A. Koshoffer, L. Newman, J. Savolainen, Advancing viewpoint merging in requirements engineering: A theoretical replication and explanatory study, Requirements Engineering 22 (2017) 317–338.

[23] M. Alenazi, N. Niu, J. Savolainen, A novel approach to tracing safety requirements and state-based design models, in: Proceedings of the 42nd International Conference on Software Engineering, ICSE'20, Seoul, South Korea, 2020, pp. 848–860.

[24] T. Bhowmik, N. Niu, J. Savolainen, A. Mahmoud, Leveraging topic modeling and part-of-speech tagging to support combinational creativity in requirements engineering, Requirements Engineering 20 (2015) 253–280.

[25] S. Reddivari, Z. Chen, N. Niu, ReCVisu: A tool for clustering-based visual exploration of requirements, in: Proceedings of the 20th IEEE International Requirements Engineering Conference, RE'12, Chicago, IL, USA, 2012, pp. 327–328.