Early Requirements Traceability with Domain-Specific Taxonomies - A Pilot Experiment

Michael Unterkalmsteiner
Department of Software Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
michael.unterkalmsteiner@bth.se

Abstract*—*Background: **Establishing traceability from requirements documents to downstream artifacts early can be beneficial as it allows engineers to reason about requirements quality (e.g. completeness, consistency, redundancy). However, creating such early traces is difficult if downstream artifacts do not exist yet.** ***Objective:*** **We propose to use domain-specific taxonomies to establish early traceability, raising the value and perceived benefits of trace links so that they are also available at later development phases, e.g. in design, testing or maintenance.** ***Method:*** **We developed a recommender system that suggests trace links from requirements to a domain-specific taxonomy based on a series of heuristics. We designed a controlled experiment to compare industry practitioners' efficiency, accuracy, consistency and confidence with and without support from the recommender.** ***Results:*** **We have piloted the experimental material with seven practitioners. The analysis of self-reported confidence suggests that the trace task itself is very challenging as both control and treatment group report low confidence on correctness and completeness.** ***Conclusions:*** **As a pilot, the experiment was successful since it provided initial feedback on the performance of the recommender, insight on the experimental material and illustrated that the collected data can be meaningfully analysed.**

***Index Terms*—Traceability, Requirements, Domain-specific Taxonomy, Recommender, Pilot Experiment**

I. Introduction

Tracing requirements to downstream artifacts has benefits, such as more efficient and correct software maintenance [1], enables requirements-based testing [2], and is often demanded by regulations on software production [3]. However, requirements engineers lack motivation to create traces as they usually are not the beneficiaries of traceability [4]. Furthermore, creating traces from requirements to downstream artifacts requires that these downstream artifacts already exist. Early requirements traces would increase their value for engineers, as they would allow to reason about requirements correctness, completeness or consistency.

To reap these early trace benefits for requirements engineers, we propose to trace requirements to domain-specific taxonomies. This enables early requirements analysis by exploiting the information encoded in a taxonomy, i.e. the definitions and hierarchies of domain concepts. For example, engineers can reason about the completeness and correctness of requirements specifications [5], [6], [7]. This is particularly useful when the number of requirements is high (in the order of thousands) and are written over a longer period of time by different engineers. Furthermore, when information systems are used where direct traces between artefacts are not possible (e.g. in an outsourcing scenario), the taxonomy serves as an "index" to establish traceability.

The approach to use a domain-specific taxonomy as a mean to enable early traceability across time, infrastructure and organizational borders is contingent on two assumptions: (1) such a taxonomy exists or can be created at low cost. In this paper, we assume that such a taxonomy exists; (2) engineers are able to associate taxonomy concepts to requirements. The evaluation of this second assumption is subject of an ongoing research project. The main research question we address is:

RQ To what extent can a recommender system support practitioners in associating requirements with concepts defined in a taxonomy?

We implemented a recommender system, CCR, that we are currently evaluating with practitioners. In this paper, we report on a pilot that aimed at validating the experiment design, material and instrumentation. As the number of participating subjects in the pilot was seven, we are not able yet to answer our main research question. However, these initial results provide an indication that the task, associating requirements with taxonomy objects, is difficult, even for domain experts. Furthermore, we illustrate metrics which could be helpful for researchers evaluating recommender systems in general.

The remainder of the paper is structured as follows. In Section II we provide background on knowledge organization systems, introduce the basic underpinnings of CCR and point to related work. We document the experimental plan in Section III, analyze the results in Section IV and discuss the outcome of the pilot experiment in Section V. We conclude the paper in Section VI.

II. Background and Related Work

A. Knowledge Organization Systems

There exists a wide spectrum of processes and artifacts that are used in practice to represent entities and their relationships for various knowledge oriented applications [8], [9]. A controlled vocabulary is a closed list of terms that describes a subject area. It homogenizes the assignment of terms to concepts as everyone is limited to the same, existing definitions

(for example, the Library of Congress Subject Headings[1]). Taxonomies provide additional structure to controlled vocabularies. The term "taxonomy" is rooted in the greek *taxis*, which broadly means the arrangement of things, and *nomos*, meaning law or science. A taxonomy is a set of rules to order things, abstract or concrete. Taxonomies can, but don't have to, be hierarchical and encode a particular slice of knowledge about the world. Modifications to content and structure typically require the consensus of the community that has ownership over the taxonomy. The IEEE Thesaurus[2] is an example that contains engineering, technical and scientific terms, organized in a broad to narrow progression. While taxonomies are useful constructs to encode and share knowledge, they are one-dimensional and based on a closed vocabulary [9]. Ontologies allow to model relationships between concepts, even if they are part of different taxonomies. This makes ontologies versatile in representing domain knowledge, requiring however also formal languages, such as the Web Ontology Language (OWL)[3], that enable computational reasoning and applications in artificial intelligence [9].

In the remainder of this paper, we focus on the use of taxonomies as knowledge organization systems since we studied the proposed approach, described next, in a context in which a domain-specific taxonomy already exists. Nevertheless, we point out that the basic idea we propose is independent from the underlying knowledge structure. The type of analyses, however, that can be performed once requirements are traced, depends on the sophistication of the used knowledge organization system. For example, requirements mapped to a controlled vocabulary may allow analyses on the consistency of requirements. Analyses that are targeted at understanding the completeness of requirements specifications may require ontologies that are able to reflect structure and relationships between entities.

B. The CC Recommender (CCR)

CoClass[4] (CC) is a taxonomy that describes objects in the construction domain. For our approach, we make the assumption that the most information-bearing language construct is the noun. Therefore, to establish traces between requirements and taxonomy objects, it would make sense to base those traces on nouns. We use a basic natural language processing pipeline that consists of a segmenter, tokenizer, stemmer and part-of-speech tagger to identify nouns, using the DKPro framework [10]. The domain-specific terminology found in the taxonomy and in the requirements uses agglutination, i.e. complex terms are built from two or more component morphemes. Therefore, we also de-compound the identified nouns with SECOS [11].

Once the nouns in a requirement are identified, we associate each noun with $0..n$ taxonomy objects. This association is established by three predictors. Each predictor score contributes to a confidence score $[0..1]$ that is used to rank the taxonomy objects. This strategy allows us to add new predictors in the future and to weigh the components contributing to the total score. Next, we describe the currently implemented predictors that are calculated for each noun found in a requirement.

a) Exact match predictor: If a stemmed, de-compound noun is found in the requirement and in the CC taxonomy, the score is computed as follows:

$$P_{exact} = \frac{1}{f_{noun}}$$

where f_{noun} refers to the number of taxonomy objects in which the noun appears in. The more prevalent the noun is, i.e. the less distinguishing power between objects it has, the lower the predictive score.

b) Semantic similarity predictor: While the requirements are written by domain experts, they are not necessarily using the exact terminology that is used in the taxonomy. We developed therefore a predictor using word embeddings [12]that exploits semantic relatedness among nouns. Instead of using a pre-trained model, e.g. from Wikipedia articles, we trained our own domain-specific model. First, we constructed a text corpus by searching the web programmatically[5] for nouns used in the labels of CC taxonomy objects. This resulted in 540,409 documents from which we extracted[6] the text to construct a word2vec model[7]. Then, for each noun in a requirement, we use the model to find the 10 most similar nouns, i.e. "proxies", and try to find them in the set of taxonomy nouns. Any such identified "proxy" produces another association between a requirement noun and a taxonomy object, with the score:

$$P_{similarity} = \frac{1}{f_{proxy} * \cos(\theta_{noun-proxy})}$$

where f_{proxy} refers to the number of taxonomy objects in which the "proxy" appears and $\cos(\theta_{noun-proxy})$ refers to the cosine similarity of noun and "proxy" based on the custom word2vec model. The more similar a "proxy" is to a noun found in a requirement, and the less frequently it appears in the taxonomy, the higher the semantic similarity predictor score.

c) History predictor: Finally, we take into consideration past decisions, that is, data reflecting whether an association between a particular noun and taxonomy object was accepted or rejected by the user of the recommender. After a particular noun-object association has been rejected n times (the default is five, but can be configured to any number), the predictor score is set to $-\infty$. Otherwise, the score is calculated as:

$$P_{history} = \frac{\frac{f_{assoc} - min(f_{assoc})}{max(f_{assoc}) - min(f_{assoc})}}{f_{noun}}$$

where f_{assoc} refers to the number of existing associations between the noun and taxonomy object. In the numerator of

[1]http://id.loc.gov/authorities/subjects.html
[2]https://www.ieee.org/publications/services/thesaurus.html
[3]https://www.w3.org/TR/owl2-overview/
[4]https://coclass.byggtjanst.se/about#about-coclass
[5]https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/
[6]https://textract.readthedocs.io
[7]https://radimrehurek.com/gensim/models/word2vec.html

the fraction we scale the frequency of occurrences to $(0..1]$. All predictors produce scores in this range, which allows us to calculate an overall confidence score:

$$P_{confidence} = \frac{P_{exact} + P_{similarity} + P_{history}}{3}$$

We have implemented CCR using INCEpTION [13], a web-based annotation platform. The user is presented a requirement and for each (recognized) noun, one or more suggested associations with taxonomy objects are shown. The suggestions are ordered by the calculated $P_{confidence}$ score. The user can then either reject or accept suggestions until no more suggestions are available. The source code for CCR, the instrumentation of the experiment, and the collected data together with the statistical analysis is available online[8].

C. Related Work

We direct readers to the systematic literature review by Dermeval et al. [14], who reviewed the research on ontologies as knowledge organization systems supporting requirements engineering activities, and to Borg et al. [15] who reviewed information retrieval approaches to traceability recovery.

III. Experiment Planning

We have designed a quasi-experiment with industry practitioners, following the guidelines by Wohlin et al. [16] and report the experiment planning according to Jedlitschka et al. [17]. We decided against a randomized design as we wanted to balance control and treatment group with respect to the participants' experience on requirements and the domain-specific taxonomy. Furthermore, we decided to involve industry practitioners to avoid constructing artificial domain-specific material (requirements, taxonomy) that would be suitable for e.g. student subjects.

A. GQM

We refine the research question posed in the introduction with the Goal-Question-Metric (GQM) approach [18].

a) Goal: Analyze manual and recommender aided association of requirements with a taxonomy, for the purpose of evaluation with respect to efficiency, accuracy, consistency and confidence from the viewpoint of domain experts in the context of an infrastructure project.

The evaluation aspects of efficiency, accuracy, consistency and reliability were inspired by work on effectiveness evaluation of expert systems [19].

b) Questions:

- Q1 Is there a difference in time spent to create manual and recommender aided associations?
- Q2 Is there a difference in accuracy between manual and recommender aided associations?
- Q3 Is there a difference in consistency between manual and recommender aided associations?
- Q4 Is there a difference in reported confidence by engineers creating manual and recommender aided associations?

[8]https://zenodo.org/record/3827169

c) Metrics: Table II maps metrics to questions.

TABLE II: Metrics to answer questions

Question	Metrics
Q1	time spent per requirement (M1)
Q2	expert-based judgment on correctness of associations (M2)
Q3	within-group variation of made associations (M3)
Q4	self-reported confidence in terms of completeness (M4) and correctness (M5) of made associations

We measure M1 by the time in seconds spent to associate a requirement with zero or more objects from the taxonomy. The correctness of associations (M2) is assessed independently by two domain experts. They are tasked to distribute 10 points to the instances of associations made by the experiment participants. The judgment is made without knowledge of whether the associations were made with or without the aid of the recommender. We measure thereby the relative and not absolute correctness of the associations. We measure variance within the group (M3) by encoding each requirement as a vector that represents the associated taxonomy objects. The larger the angle between the vectors representing the association instances, the larger the within group variation (and the lower the association consistency). The idea for this measure stems from the vector space model [20] that is often used to encode text documents to analyze their similarity. Finally, we measure confidence in completeness (M4), i.e. whether all relevant taxonomy objects were associated with the requirement, and correctness (M5), i.e. whether the made associations are correct. The experiment participants self-report their confidence on a scale from -2 to +2 per requirement.

B. Hypotheses and Variables

Based on the metrics defined in the GQM, we formulate five hypotheses pairs. We show one pair and explain next how the five pairs are generated.

$$\begin{aligned} H_{0n} &: Mn_{CCR} = Mn_{search} \\ H_{1n} &: Mn_{CCR} \neq Mn_{search} \end{aligned} \tag{1}$$

where $n = [1..5]$ and the dependent variables $M_{n=[1..5]}$ refer to the metrics defined in Table II. The factor is how the association between requirement and CC object is supported in INCEpTION, with two possible treatments. CCR refers to the treatment using the CC recommender we described in Section II-B, while $search$ refers to the treatment using the full-text search to find CC taxonomy objects in INCEpTION.

Note that our hypothesis formulations do not assume directionality because we do not compare an established and a new method. Both alternatives, associating requirements to CC objects with and without CCR support, are activities that are not familiar to the participants (control of experience is discussed in Section III-C).

C. Participants

We recruited seven domain experts with varying experience working with requirements and the domain-specific taxonomy.

TABLE I: Pre-questionnaire results

Variables	CCR				$search$		
	P1	P2	P3	P4	P5	P6	P7
Current role	Product owner asset management	Contract specialist	Proj. manager technical requirements	Information management research	Proj. manager technical requirements	Bridge specialist	Proj. manager technical requirements
Years in role	5	10	6	2	5	9	6
Total exp.	5	10	23	15	25	35	23
Writing requirements	once a month	a couple of times per year	a couple of times per year	a couple of times per year	a couple of times per year	never	a couple of times per year
Read requirements	once a month	daily	once a week	a couple of times per year	daily	once a week	once a week
Experience CC	yes	no	yes	yes	yes	no	no
Use of CC	once a month	N/A	a couple of times per year	a couple of times per year	daily	N/A	N/A
Location	onsite	offsite	onsite	onsite	onsite	offsite	onsite

Available frequency options: daily, once a week, once a month, a couple of times per year, less frequently, never

The participants filled in a questionnaire that we used balance treatment and control group, based on experience with reading and writing requirements, use of the CC taxonomy and their overall experience in the construction domain (Table I). Two remotely participating subjects were equally distributed between treatment and control group. There was an uneven number of participants and we chose to add four to the CCR treatment in order to collect more usage experience on the instrument under investigation.

D. Materials

We randomly sampled 100 from a set of 1,216 requirements that belong to an ongoing infrastructure project. The average number of words per requirements was 19 (minimum: 5, maximum: 77). The CC taxonomy contained 1,420 objects. Each object has a label, a description and associated synonyms. Finally, the participants were given a spreadsheet in which they reported the duration (M1) and confidence (M4, M5) for each requirement they mapped to objects in the CC taxonomy.

E. Tasks

The participants used the web-based annotation system INCEpTION. Both groups were tasked to annotate the same requirements, in the same order. The only difference between the two groups was that the treatment group received suggestions from CCR, which they either accepted or rejected, while the control group used the built-in search functionality of the annotation tool to find the relevant objects.

F. Experiment Design

We chose for the pilot a simple one factor and two treatments design [16]. All participants received the same requirements and were using the assigned treatment throughout the experiment. We assigned four participants to the CCR and three participants to the $search$ treatment (see Table I).

G. Procedure

We carried the pilot experiment out on January 15, 2020. We allocated one hour for explaining the principle idea of the activity, associating requirements to CC objects, and illustrated the mechanics of the annotation task with INCEpTION, both using the CCR and the built-in search functionality. The remainder of the time (two hours) was allocated to perform the experimental task. All participants, except two who were connected via a videoconferencing software, were located in the same room. While the participants performed the tasks, the author of this paper answered questions and helped participants in case of technical issues.

H. Analysis Procedure

Due to the low number of observations collected in the pilot experiment, we resolved to a non-parametric test statistic with lower power (Mann-Whitney-Wilcoxon) rather than its parametric counterpart (t-test). While the results are therefore less robust, we deem it important to also pilot the analysis procedure, especially since the raw data required intermediate analyses to evaluate accuracy (Section IV-B) and consistency (Section IV-C), as described in the respective sections.

I. Deviations from the Plan

We spent 1.75 hours, instead of the allocated 1 hour, on the experiment introduction. Including a 15 minute break, 1 hour was spent on the experiment execution, instead of the originally planned 2 hours.

IV. Analysis

During the allocated time for the experiment execution, the participants completed a varying number of tasks (requirements), shown in Table III. We limit therefore our analysis to the requirements that were annotated by all participants ($n_{requirements} = 7$). Next, we analyse the results, answering our four GQM questions.

TABLE III: Finished tasks by participants P1-P7

	CCR				$search$		
	P1	P2	P3	P4	P5	P6	P7
Number of tasks	28	32	24	14	12	13	7
Median time per task (s)	72	59	69	179	105	17	128

A. Efficiency

Median duration in groups CCR and $search$ was 61 and 101 seconds. The distribution in the two groups did not differ significantly (Mann-Whitney-Wilcoxon $U = 209$, $n_{CCR} = 28$, $n_{search} = 21$, $p = 0.09$). We cannot reject H_{01} at $\alpha < 0.05$. Figure 1 shows a box plot of the annotation duration for each requirement.

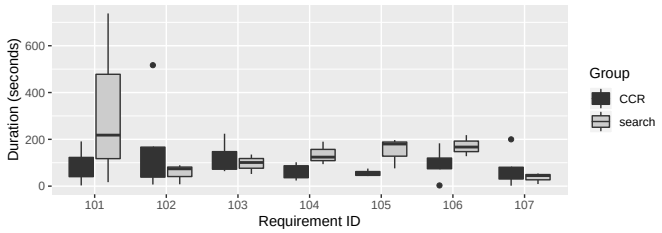

Fig. 1: Annotation duration (seconds) per requirement

B. Accuracy

The participants produced 49 observations, i.e. associations between requirements and CC objects. Two domain experts evaluated their relative accuracy (see Section III-A) by distributing 10 points on the associations for each requirement. Table IV shows the frequency of the evaluators agreements/disagreements and the score they were apart, indicating that the two experts had a good agreement.

TABLE IV: Inter-rater agreement

Agreements (score difference = 0)	23
Disagreements (score difference = 1)	17
Disagreements (score difference = 2)	6
Disagreements (score difference = 3)	3
Disagreements (score difference > 3)	0

The median accuracy score in groups CCR and $search$ was 4 and 8. The distribution in the two groups differed significantly (Mann-Whitney-Wilcoxon $U = 0$, $n_{CCR} = n_{search} = 7$, $p = 0.002$). We reject H_{02} at $\alpha < 0.05$. Looking at the bar plot in Figure 2, which shows the average score of the two evaluators, we see that $search$ resulted in more accurate results than CCR.

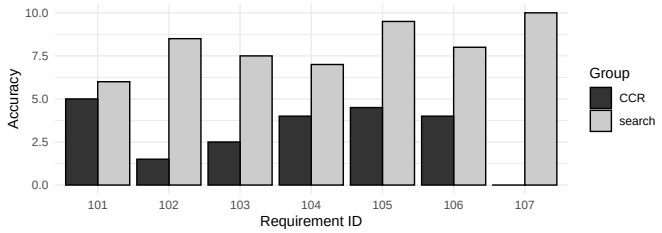

Fig. 2: Association accuracy per requirement

TABLE V: Example of coding to assess consistency

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	1	1	1	1	1	1	1	2	1	3
P2	1	1	1	1	1	1	1	5	1	1
P3	1	1	1	1	1	1	1	5	1	1
P4	1	1	1	1	1	1	1	5	1	3

C. Consistency

Table V shows an example of how four participants annotated a requirement. Each term $T_{1..10}$ is coded with a label representing a CC object (1 representing no object). For example, participant P1 associated term T8 with object 2 while P2, P3 and P4 associated the same term with object 5. In order to assess consistency between participants within one treatment group, we calculated the average pairwise cosine similarity between their association vectors which results in a score between $(0..1]$, 1 indicating complete consistency. The median consistency score in groups CCR and $search$ was 0.98 and 0.90. The distribution in the two groups did not differ significantly (Mann-Whitney-Wilcoxon $U = 34$, $n_{CCR} = n_{search} = 7$, $p = 0.25$). We cannot reject H_{03} at $\alpha < 0.05$. Figure 3 illustrates the results.

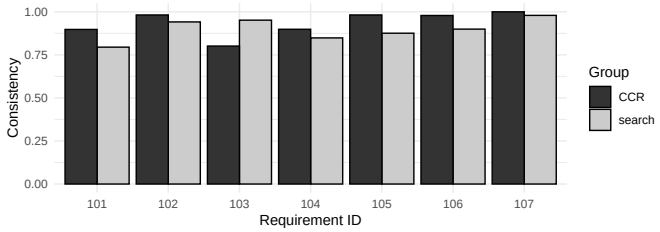

Fig. 3: Consistency per requirement

D. Confidence

Figures 4 and 5 show the results of the participants' self-reported confidence ($[-2, +2]$) in terms of correct and complete associations per requirement.

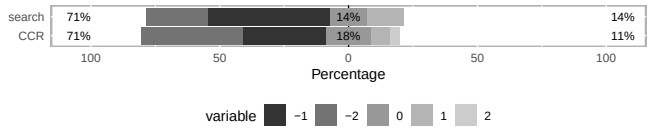

Fig. 4: Correctness confidence

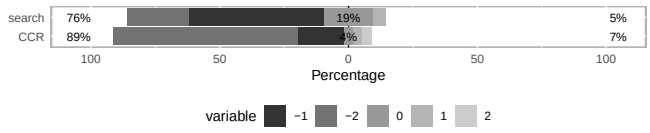

Fig. 5: Completeness confidence

Correctness confidence in groups CCR and $search$ is low (71%), neutral (18% and 14%) and high (11% and 14%). The distribution in the two groups did not differ significantly (Mann-Whitney-Wilcoxon $U = 260$, $n_{CCR} = 28$, $n_{search} = 21$, $p = 0.47$). We cannot reject H_{04} at $\alpha < 0.05$.

Completeness confidence in groups CCR and $search$ is low (89% and 76%), neutral (4% and 19%) and high (7% and 5%). The distribution in the two groups did differ significantly (Mann-Whitney-Wilcoxon $U = 162$, $n_{CCR} = 28$, $n_{search} = 21$, $p = 0.004$). We reject H_{05} at $\alpha < 0.05$, and accept that the self-reported confidence in terms of completeness is higher with $search$.

V. Discussion

The feedback collected during the experiment and the results on the self-reported confidence on the correctness and completeness of the associations between requirements and CC taxonomy objects indicate that the task is challenging, even for engineers with extensive domain experience. Both approaches, CCR and $search$ lead to low confidence on the created traces. While there are indications that traces can be created faster with CCR, $search$ traces were judged as more accurate. We received feedback during the experiment that the CCR suggested associations do not consider the context of the requirement. This corresponds well with the implementation of the used predictors which consider currently only single nouns.

A. Threats to Validity

We limit our discussion to two major threat dimensions [16].

a) Conclusion: The concept of tracing a requirement to a taxonomy was new to all participants. The training was very limited and the participants may have interpreted the task in different ways.

b) Internal: The annotation UI within INCEpTION, where suggestions are accepted with a single click and rejected with a double click, caused some confusion and the participants perceived it as error prone. Furthermore, the manual collection of spent time and confidence is error prone. Finally, the participants may have influenced each others answers by working in the same room, on the same requirements at the same time.

VI. Conclusions

We proposed early requirements tracing to domain-specific taxonomies to support the analysis of requirements specifications. We developed a recommender that suggests associations between requirements and a taxonomy in the construction domain. To evaluate the feasibility of creating such traces, we designed a controlled experiment, comparing the recommender with manually establishing associations. We measured multiple dimensions to better understand the differences between the approaches. As a pilot, the experiment was successful since it provided initial feedback on the performance of the recommender, insight on the experimental material and illustrated that the collected data can be meaningfully analysed. Future experiments can also consider the factors participant experience and the length of requirements. Furthermore, the idea of using a taxonomy as a mediator to establish trace links needs to be further validated on other artefacts than requirements, such a design documentation or source code.

Acknowledgments

The author would like to thank Claes Wohlin for providing feedback on the experiment design. This work was funded by Trafikverket (FoI KREDA).

References

[1] P. Mäder and A. Egyed, "Do developers benefit from requirements traceability when evolving and maintaining a software system?" *Empirical Software Engineering*, vol. 20, no. 2, pp. 413–441, Apr. 2015.

[2] E. Bouillon, P. Mäder, and I. Philippow, "A Survey on Usage Scenarios for Requirements Traceability in Practice," in *Req. Eng.: Foundation for Software Quality*. Essen, Germany: Springer, 2013, pp. 158–173.

[3] G. Regan, F. McCaffery, K. McDaid, and D. Flood, "Traceability-Why Do It?" in *12th Intl. Conference on Software Process Improvement and Capability Determination*. Palma, Spain: Springer, 2012, pp. 161–172.

[4] P. Arkley and S. Riddle, "Overcoming the traceability benefit problem," in *13th Intl. Req. Engineering Conference*, Aug. 2005, pp. 385–389.

[5] D. V. Dzung and A. Ohnishi, "Improvement of Quality of Software Requirements with Requirements Ontology," in *9th Intl. Conference on Quality Software*, Aug. 2009, pp. 284–289.

[6] L. Kof, R. Gacitua, M. Rouncefield, and P. Sawyer, "Ontology and Model Alignment as a Means for Requirements Validation," in *4th Intl. Conference on Semantic Computing*, Sep. 2010, pp. 46–51.

[7] T. Moser, D. Winkler, M. Heindl, and S. Biffl, "Requirements Management with Semantic Technology: An Empirical Study on Automated Requirements Categorization and Conflict Analysis," in *23rd Intl. Conference on Advanced Information Systems Engineering*. London, UK: Springer, 2011, pp. 3–17.

[8] M. Gruninger, O. Bodenreider, F. Olken, L. Obrst, and P. Yim, "Ontology Summit 2007–Ontology, taxonomy, folksonomy: Understanding the distinctions," *Applied Ontology*, vol. 3, no. 3, pp. 191–200, 2008.

[9] L. M. Garshol, "Metadata? Thesauri? Taxonomies? Topic Maps! Making Sense of it all," *Journal of Information Science*, vol. 30, no. 4, pp. 378–391, Aug. 2004.

[10] R. Eckart de Castilho and I. Gurevych, "A broad-coverage collection of portable NLP components for building shareable analysis pipelines," in *Workshop on Open Infrastructures and Analysis Frameworks for HLT*. Dublin, Ireland: Assoc. for Comp. Linguistics, Aug. 2014, pp. 1–11.

[11] C. B. Martin Riedl, "Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods," in *15th Annual Conference of the North American Chapter of the Assoc. for Comp. Linguistics: Human Language Technologie*, San Diego, CA, USA, 2016, pp. 617–622.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *preprint arXiv:1301.3781*, 2013.

[13] J.-C. Klie, M. Bugert, B. Boullosa, R. E. d. Castilho, and I. Gurevych, "The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation," in *27th Intl. Conference on Computational Linguistics: System Demonstrations*. Assoc. for Comp. Linguistics, Jun. 2018, pp. 5–9.

[14] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva, "Applications of ontologies in requirements engineering: a systematic review of the literature," *Requirements Engineering*, vol. 21, no. 4, pp. 405–437, Nov. 2016.

[15] M. Borg, P. Runeson, and A. Ardö, "Recovering from a decade: a systematic mapping of information retrieval approaches to software traceability," *Empirical Software Engineering*, vol. 19, no. 6, pp. 1565–1616, 2014.

[16] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering: an introduction*. Norwell: Kluwer Academic Publishers, 2000.

[17] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, "Reporting Experiments in Software Engineering," in *Guide to Advanced Empirical Software Engineering*. London: Springer, 2008, pp. 201–228.

[18] V. R. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[19] R. Sharda, S. H. Barr, and J. C. McDonnell, "Decision Support System Effectiveness: A Review and an Empirical Test," *Management Science*, vol. 34, no. 2, pp. 139–159, Feb. 1988.

[20] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.