Requirements Engineering
https://doi.org/10.1007/s00766-022-00371-x

ORIGINAL ARTICLE

Causality in requirements artifacts: prevalence, detection, and impact

Julian Frattini[1] · Jannik Fischbach[2,3] · Daniel Mendez[1,4] · Michael Unterkalmsteiner[1] · Andreas Vogelsang[3] · Krzysztof Wnuk[1]

Received: 23 June 2021 / Accepted: 13 December 2021

Abstract

Causal relations in natural language (NL) requirements convey strong, semantic information. Automatically extracting such causal information enables multiple use cases, such as test case generation, but it also requires to reliably detect causal relations in the first place. Currently, this is still a cumbersome task as causality in NL requirements is still barely understood and, thus, barely detectable. In our empirically informed research, we aim at better understanding the notion of causality and supporting the automatic extraction of causal relations in NL requirements. In a first case study, we investigate 14.983 sentences from 53 requirements documents to understand the extent and form in which causality occurs. Second, we present and evaluate a tool-supported approach, called CiRA, for causality detection. We conclude with a second case study where we demonstrate the applicability of our tool and investigate the impact of causality on NL requirements. The first case study shows that causality constitutes around 28 % of all NL requirements sentences. We then demonstrate that our detection tool achieves a macro-F_1 score of 82 % on real-world data and that it outperforms related approaches with an average gain of 11.06 % in macro-Recall and 11.43 % in macro-Precision. Finally, our second case study corroborates the positive correlations of causality with features of NL requirements. The results strengthen our confidence in the eligibility of causal relations for downstream reuse, while our tool and publicly available data constitute a first step in the ongoing endeavors of utilizing causality in RE and beyond.

Keywords Causality · Multi-case study · Requirements engineering · Natural language processing

1 Introduction

Motivation Sentences containing causal relations, for example "A confirmation message shall be shown if the system has successfully processed the data.", are often used to capture the intended behavior of a system. In fact, causal relations are inherently embedded in many textual descriptions of requirements. Both understanding the extent of use, but also detecting and reliably extracting these causal relations, provide great potential for applications in the domain of Requirements Engineering (RE). Among these are, for instance, supporting the automated test case generation [16, 17] or facilitating reasoning about inter-requirements dependencies [15]. However, the automated extraction of causal relations from requirements is still challenging for two reasons: On the one hand, even though controlled natural language in RE [20, 35] aims to minimize ambiguity and can easily be reused for further formalization [23, 46], unrestricted natural language is still predominantly used in RE [50]. This complicates information retrieval from

✉ Julian Frattini
julian.frattini@bth.se

Jannik Fischbach
jannik.fischbach@qualicen.de

Daniel Mendez
daniel.mendez@bth.se

Michael Unterkalmsteiner
michael.unterkalmsteiner@bth.se

Andreas Vogelsang
vogelsang@cs.uni-koeln.de

Krzysztof Wnuk
krzysztof.wnuk@bth.se

[1] Blekinge Institute of Technology, Karlskrona, Sweden

[2] Qualicen GmbH, München, Germany

[3] University of Cologne, Cologne, Germany

[4] Fortiss GmbH, Munich, Germany

Published online: 09 February 2022

requirements due to the inherent complexity and ambiguity of NL. On the other hand, causal relations can occur in different forms [3] such as *marked*/*unmarked* (i.e., containing a cue phrase indicating the causal relationship) or *explicit*/*implicit* (i.e., explicitly stating both the cause and the effect), further rendering the identification and extraction of causes and effects cumbersome. Existing approaches still fail to extract causal relations from NL with a performance that allows for efficient and reliable use in practice [1]. We therefore argue that a novel method for detecting and extracting causal relations from requirements is imperative for the effective utilization of causality in RE.

Contribution Causality extraction entails two distinct challenges: first, one needs to determine whether a requirement contains causal relations. Only sentences containing causal relations are eligible for extraction, so sentences containing no causal relations can be discarded. Second—if they contain causal relations, these need to be properly understood and extracted. Addressing both challenges requires comprehending to which extent, form, and complexity causality occurs in practice in RE. Reliable knowledge about the distribution of causality is a necessary precondition to develop efficient approaches for the automated detection and extraction of causal relations. However, empirical evidence on causality in requirements artifacts is still unavailable to this day. In this manuscript, we report on how we addressed those challenges to close the existing research gap by making the following contributions (C):

C 1 *Prevalence of Causality*: We report on an exploratory case study analyzing the extent, form, and complexity of causality in requirements rooted in 14,983 sentences and emerging from 53 requirement documents, which originate from 18 different domains. We corroborate that causality tends to occur predominantly in *explicit* and *marked* form, and that about 28 % of the analyzed sentences contain causal information about the expected system behavior. This strengthens our confidence in the relevance of causality and, in consequence, of our approach to automatically extract causality.

C 2 *Automated Detection of Causality*: We present our tool-supported approach CiRA (**C**ausality **i**n **R**equirement **A**rtifacts), which constitutes a first step toward causality extraction from NL requirements by automatically detecting causal relations in NL requirements. We train and empirically evaluate CiRA using the pre-analyzed data set and achieve a macro-F_1 score of 82 %. Compared to baseline systems that classify causality relying on the presence of certain cue phrases, or shallow ML models, CiRA leads to an average performance gain of 11.43 % in macro-Precision and 11.06 % in macro-Recall.

C 3 *Impact of Causality*: We report on a second exploratory case study evaluating the correlation between the occurrence of causality and its effects on the requirements life cycle, not only demonstrating a possible use case of the automatic causality detection approach and tool but also corroborating the positive impact on causality on the requirements process.

C 4 *Open Data and Source Code*: To strengthen the transparency and, thus, the credibility of our research, but also to facilitate independent replications, we publicly disclose our tool, code, and data set used in the case study. A demo of CiRA can be accessed at http://www.cira.bth.se/bert. Our code and annotated data sets can be found at https://doi.org/10.5281/zenodo.5596668.

1.1 Previously published material

This manuscript extends our previously published conference paper [14] in the following aspects: we extend our case study (C1) and development of our own approach (C2) by the aforementioned second case study (C3) based on an extensive data set of requirements from a multinational software development company. In addition, we expand the evaluation of the resulting data from the first case study in response to discussions with the requirements engineering community to increase the generalizability of our results. Please note that we took the liberty of intentionally reusing minor parts of our previously published material for this manuscript at hand in a verbatim manner, such as discussions on related work or terminological definitions.

1.2 Outline

The manuscript is structured as follows: Sect. 2 introduces the terminology used throughout the manuscript based on the established literature. Sect. 3 reports on the first case study investigating the extent, form, and complexity to which causality is used in NL requirements (C1). Our approach for automatically detecting causal requirements (C2) is presented in Sect. 4. Sect. 5 reports on the second case study (C3) investigating the impact of causality in natural language requirements on their life cycle. Section 6, finally, presents related work in the research area before concluding our work with Sect. 7.

2 Terminology

The concept of causality has received notable attention in the studies of various disciplines, e.g., in psychology [53]. Before investigating the extent to which causality occurs in requirements, we elaborate on a definition of causality.

Concept of Causality Causality describes a relation between at least two events: a causing event (the *cause*) and a caused event (the *effect*). An event is commonly defined as "any situation (including a process or state) that happens or occurs either instantaneously (punctual) or during a period of time (durative)" [39]. The connection between causes and effects is counterfactual [34]: if a cause c_1 does not occur, then an effect e_1 cannot occur either. Consequently, a causal relation entails that the effect may only occur *if and only if* the cause has occurred. If this is not the case, then the relation might be confounded and is hence not causal. This relation can be interpreted in the view of Boolean algebra as an equivalence between a cause and effect ($c_1 \Leftrightarrow e_1$): if the cause is true, the effect is true and if the cause is false, the effect is also false. The representation of a causal relationship as a logical equivalence is not entirely reflecting the nature of the relation, especially in regards to the temporal order of events, which is not determined in propositional logic. The challenges of formalizing causal relationships both regarding the used notation and the ambiguity when interpreting these relationships are discussed in depth in a different paper [13]. For the remainder of this manuscript, we use the notation of a logical equivalence to represent a causal relationship. We refer the reader interested in an extended discussion on the logical formalization of causal relationships to our previous publication on the matter [13]. The type of causal relation between the two events can take one of three different forms [52]: a *causing*, *enabling*, or *preventing* relationship.

- c_1 *causes* e_1: If c_1 occurs, e_1 also occurs ($c_1 \Leftrightarrow e_1$). This can be illustrated by REQ 1: "After the user enters a wrong password, a warning window shall be shown." In this case, the wrong input is the trigger to display the window.
- c_1 *enables* e_1: If c_1 does not occur, e_1 does not occur either (e_1 is not enabled, ($\neg c_1 \Leftrightarrow \neg e_1$)). REQ 2: "As long as you are a student, you are allowed to use the sport facilities of the university." Only the student status enables to do sports on campus.
- c_1 *prevents* e_1: If c_1 occurs, e_1 does not occur ($c_1 \Leftrightarrow \neg e_1$). REQ 3: "Data redundancy is required to prevent a single failure from causing the loss of collected data." There will be no data loss due to data redundancy.

Temporal Ordering of Causes and Effects Causes and effects can be related to each other in three temporal ways [39]. In the first temporal relation, the cause occurs *before* the effect (*before* relation). In REQ 1, the user has to enter a wrong password before the warning window will be displayed. In this example, the cause and effect represent two punctual events. In the second temporal relation, the occurrence of the cause and effect *overlaps*: "The fire is burning down the house." In this case, the occurrence of the emerging fire overlaps with the occurrence of the increasingly brittle house (*overlaps* relation). In the third temporal relation (*during* relation), cause and effect occur simultaneously. REQ 2 describes such a relation: the effect—being allowed to do sports on the campus—is only valid as long as one has the student status. The start and end time of the cause is therefore also the start and end of the effect. Here, both events are durative.

Forms of Causality The form in which causality can be expressed has three further characteristics [3]: *marked* and *unmarked* causality, *explicit* and *implicit* causality, and *ambiguous* and *non-ambiguous* regarding its cue phrases, a linguistic concept commonly used when dealing with causality in natural language [5, 22]. A cue phrase is defined as "a word, a phrase, or a word pattern, which connects one event to the other with some relation" [5] and therefore a lexical indicator for the causal relationship.

- *Marked and unmarked*: A causal relation is *marked* if a certain cue phrase indicates its causality. The requirement "If the user presses the button, a window appears" is *marked* by the cue phrase "if", while "The user has no admin rights. He cannot open the folder." is *unmarked*.
- *Explicit and implicit*: An *explicit* causal relation contains information about both the cause and effect. The requirement "In case of an error, the systems prints an error message to the console" is *explicit* since it contains both the cause (error) and effect (error message). "A parent process kills a child process" is *implicit* because the effect that the child process is terminated is not explicitly stated. Implicitly causal sentences are particularly hard to process and might be a potential source of ambiguity in RE due to their obscuring nature.
- *Ambiguous and non-ambiguous cue phrases*: Due to the specificity of most cue phrases in *marked* causality, it seems feasible to deduce the classification of a sentence as containing causality based on the occurrence of certain cue phrases. However, certain cue phrases (e.g., since) indicate causality, but also occur in other contexts (e.g., denoting time constraints). Such cue phrases are called *ambiguous*, while cue phrases (e.g., because) that predominantly indicate causality are called *non-ambiguous*.

Complexity of Causality All previous examples use the complexity-wise simplest form of causality where the causal relation consists of one single cause and one single effect. Due to the increasing complexity of systems, however, the expected behavior is described by multiple causes and effects that are connected to each other. They can be linked either by conjunctions ($c_1 \wedge c_2 \wedge \cdots \Leftrightarrow e_1$) or disjunctions ($c_1 \vee c_2 \vee \cdots \Leftrightarrow e_1$) or a combination of both. Furthermore, the constituents of causal relations can be contained in more than one sentence, which is a significant challenge for causality extraction as it increases the scope of causality detection beyond one single sentence. Therefore, we also consider *two-sentence causality*. However, causal relations scattered over more than two adjacent sentences are not considered in this research work. Additionally, the complexity increases when several causal relations are linked together, i.e., if the effect of a relation r_1 represents a cause in another relation r_2. We define such causal relations, where r_2 is dependent on r_1, as *event chains* (e.g., $r_1 : c_1 \Leftrightarrow e_1$ and $r_2 : e_1 \Leftrightarrow e_2$).

3 Case study 1: prevalence of causality in requirement artifacts

We designed and conducted the case study following the well-established guidelines of Runeson and Höst [45]. Our case study is of exploratory nature based on the classification of Robson [44], as we aim to unravel new insights into causality in requirement artifacts. In this section, we describe our *research questions*, *study objects*, *study design*, *study results*, and *threats to validity*. We conclude by giving an overview of the *implications of the study* on causality detection and extraction from NL requirements.

3.1 Research questions

Our goal in understanding the prevalence of causality in requirements artifacts encompasses the extent, form, and complexity of causality. Based on the terminology previously established in Sect. 2, we investigate the following research questions (RQ) in the scope of this first case study:

RQ 1 To which degree does causality occur in requirement documents?

RQ 2 How often do the relations *cause*, *enable*, and *prevent* occur?

RQ 3 How often do the temporal relations *before*, *overlap*, and *during* occur?

RQ 4 In which form does causality occur in requirement documents?

- RQ 4a: How often does *marked* and *unmarked* causality occur?
- RQ 4b: How often does *explicit* and *implicit* causality occur?
- RQ 4c: Which causal cue phrases are used? Are they mainly *ambiguous* or *non-ambiguous*?

RQ 5 At which complexity does causality occur in RE documents?

- RQ 5a: How often do multiple causes occur?
- RQ 5b: How often do multiple effects occur?
- RQ 5c: How often does *two-sentence causality* occur?
- RQ 5d: How often do *event chains* occur?

RQ 6 Is the distribution of labels in all categories domain-independent?

3.2 Study objects

To obtain evidence on the extent to which causality is used in requirements artifacts in practice, we had to generate a large and representative collection of artifacts. We considered data sets as eligible for our case study based on three criteria: (1) the data set shall contain requirements artifacts that are/were used in practice, (2) the data set shall not be domain-specific, but rather contain artifacts from different domains, and (3) the documents shall originate from a time frame of at least 10 years. Following these criteria ensures that our analysis is not restricted to a single year or domain, but rather allows for a comprehensive and generalizable view on causality in requirements. We accordingly selected the data set provided by Fischbach et al. [15]. To the best of our knowledge, this data set is currently the most extensive collection of requirements available to the research community. From its 463 documents containing 212k extracted and pre-processed sentences, we randomly selected 53 documents from the data set for our analysis. Our final data set consists of 14,983 sentences from 18 different domains (see Fig. 1).

3.3 Study design

Model the phenomenon Answering the research questions in the scope of our first case study entails to annotate the sentences of our data set with respect to the categories elicited in Sect. 2. For example, each causal sentence has to be classified in the category *Explicit* as either *explicit* or *implicit*. According to Pustejovsky and Stubbs [43], the first step in each annotation process is to "model the phenomenon" that needs to be annotated. Specifically, the phenomenon should be defined as a model *M* that consists of a vocabulary *T*, the relations *R* between the terms as well as the interpretations *I* of terms. RQ 1 can be understood as a binary annotation problem, which can be modeled as:

- **T**: {sentence, causal, not causal}
- **R**: {sentence ::= causal | not causal}

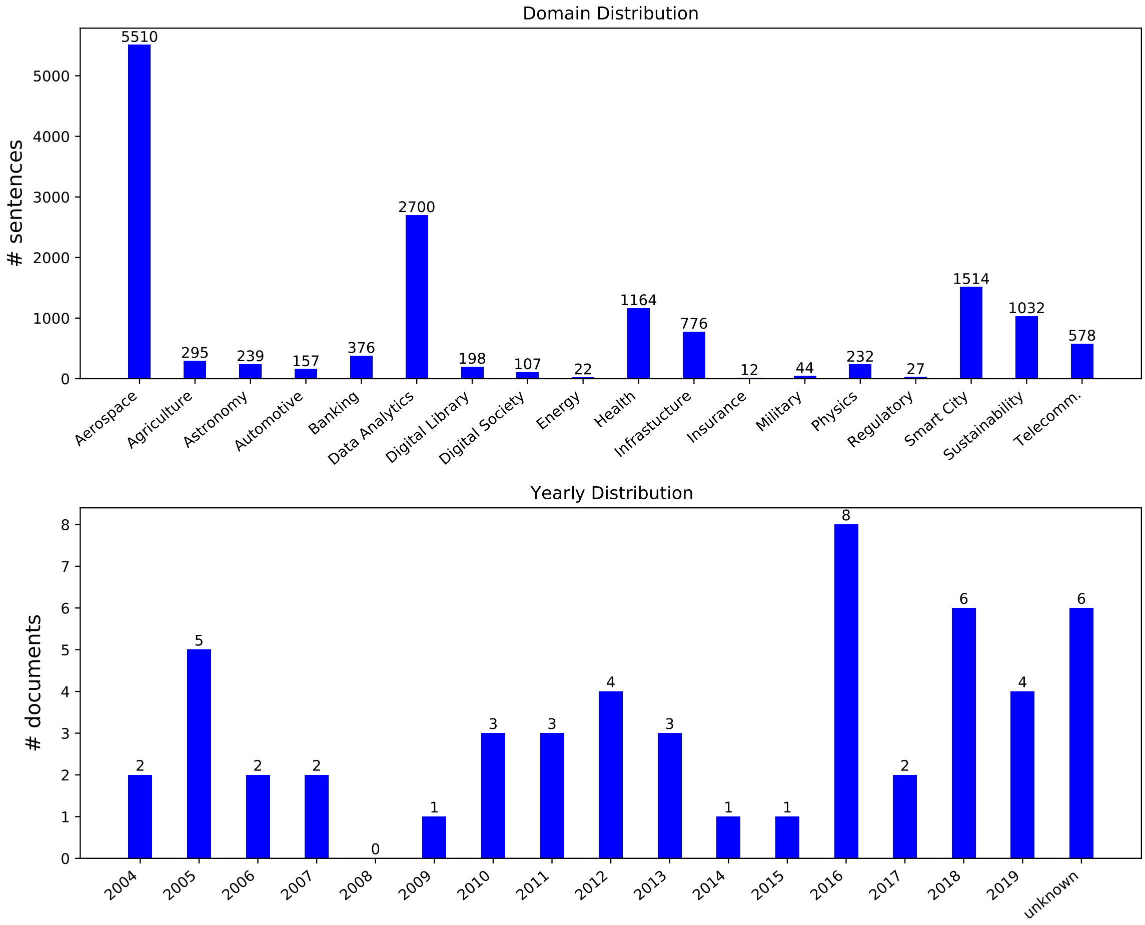


Fig. 1 Descriptive statistics of our data set. The upper graph shows the number of sentences per domain. The lower graph depicts the year of creation per document

- **I**: {causal = "A sentence is causal if it contains a relation between at least two events, where e1 causes the occurrence of e2", ¬causal = "A sentence is not causal if it describes a state that is independent on any events"}

Explicitly modeling an annotation problem according to the aforementioned framework does not only contribute an unambiguous definition of the research problem but can also be used as a guideline for the annotators explaining the meaning of the labels. Each RQ has been modeled accordingly and communicated with all annotators. In addition to interpretation *I*, we have also provided an example for each label to avoid misunderstandings. The following nine categories emerged in the process of modeling RQ 1-5, according to which we annotated our data set: Causality, Explicit, Marked, Single Sentence, Single Cause, Single Effect, Event Chain, Relationship and Temporality. We refer to all categories except Causality as *dependent* categories, as they are dependent on the Causality label. To answer RQ 6, we perform a stratified analysis for each of the aforementioned categories using the domains as strata. Due to the imbalance of the data set in respect to the domains, the requirements sentences originate from, we formulate the following null hypothesis for each category X: *sentences from different domains have the same distribution of values in category X*.

Annotation Environment We developed our own annotation platform tailored to our research questions.[1] In contrast to other annotation platforms [40] which only show single sentences to the annotators, we also display the predecessor and successor of each sentence, which is required to determine whether the causal relation is not confined to one sentence, but extends across two (see RQ 5c). For the binary annotation problems (see RQ 1, RQ 4a, RQ 4b, RQ 5a - d), we provide two labels for each category. Cue phrases present in the sentence can be manually selected by either choosing from a list of already identified cue phrases or by adding a

[1] A read-only view of the platform can be accessed at http://clabel.diptsrv003.bth.se.

Table 1 Inter-annotator agreement statistics per category. The two categories "Relationship" and "Temporality" were jointly labeled by the first and second author and therefore do not require a reliability assessment

		Causality		Explicit		Marked		Single Sentence		Single Cause		Single Effect		Event Chain		**avg.**
		0	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	
Confusion	**0**	2034	193	24	25	1	22	12	8	41	77	63	72	450	27	
Matrix	**1**	274	499	39	411	12	464	17	462	43	338	46	318	13	9	
Agreement		84.4 %		87.2 %		93.1 %		95.0 %		76.0 %		76.4 %		92.0 %		86.3 %
Cohen's Kappa		0.579		0.358		0.023		0.464		0.261		0.362		0.27		0.331
Gwet's AC1		0.753		0.84		0.926		0.945		0.645		0.625		0.91		0.806

new cue phrase using a text input field (see RQ 4c). Since RQ 2 and RQ 3 are ternary annotation problems, the platform provides three labels for these categories.

Annotation Guideline To ensure a common understanding both of causality itself and of the respective categories, we conducted a workshop with all annotators prior to the labeling process. The results of the workshop were recorded in the form of an annotation guideline. All annotators were instructed to comply with all of the annotation rules. One important, initially counter-intuitive instruction was to not entirely depend on the occurrence of cue phrases, as this approach is prone to introducing too many False Positives. Rather than focusing on lexical or syntactic attributes, the annotation process has to be initiated by fully reading the sentence and comprehending it on a semantic level. The impact of this becomes evident when considering some examples: requirements like "If the gaseous nitrogen supply is connected to the ECS duct system, ECS shall include the capability of monitoring the oxygen content in the ducting." are easy to classify as causal due to the occurrence of the cue phrase *if* and the explicit phrasing of both the cause and the effect. Requirements containing a relative clause like "Any items or issues which will limit the options available to the platform developers should be described." are more difficult to correctly classify due to the lack of cue phrases. The semantically equivalent paraphrase "If an item or issue will limit the options available to the platform developers, the item or issue should be described." reveals the causal relation contained by the requirement. A second vital instruction was to check if the cause is really necessary for the effect to occur. Only if the existence of the cause is mandatory for the effect to happen, the relation can be deemed causal.

Annotation Validity We utilize the calculation of the inter-annotator agreement to verify the reliability of our annotations. Each of the six annotators was assigned 3,000 sentences, of which 2,500 were unique and 500 overlapping. Consequently, among the approximately 15,000 annotated sentences 3,000 were labeled by two annotators. To maximize the meaningfulness of the inter-annotator agreement, the 500 overlapping sentences were selected in batches of 100, such that every annotator had an overlap with every other annotator. Based on the overlapping sentences, we calculated the Cohen's Kappa [6] measure to evaluate how well the annotators made the same annotation decision for a given category. We chose Cohen's Kappa since it is widely used for assessing inter-rater reliability [49]. However, a number of statistical problems are known to exist with this measure [36]. In case of a high imbalance of ratings, Cohen's Kappa is low and indicates poor inter-rater reliability even if there is a high agreement between the raters (Kappa paradox [11]). Thus, the calculation of Cohen's Kappa is not meaningful in such scenarios. Consequently, studies [54] suggest that Cohen's Kappa should always be reported together with the percentage of agreement and other paradox resistant measures (e.g., Gwet's AC1 measure [25]) in order to make a valid statement about the inter-rater reliability. We involved six annotators in the creation of the corpus and assessed the inter-rater reliability on the basis of 3,000 overlapping sentences, which represent about 20 % of the total data set. We calculated all measures using the cloud-based version of AgreeStat [24]. Cohen's Kappa and Gwet's AC1 can both be interpreted using the taxonomy developed by Landis and Koch [32]: values ≤ 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41–0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement. Table 1 provides an overview of the confusion matrices and calculated agreement measures per category. The inter-rater agreement for the category Causality was calculated on the basis of all 3000 overlapping sentences. Since the other categories represent specific forms of causality, we computed their inter-rater agreement only on the sentences marked as causal. Our analysis demonstrates that the inter-rater agreement of our annotation process is reliable. Across all categories, an average percentage of agreement of 86 % was achieved. Except for the categories Single Cause and Single Effect, all categories show a percentage of agreement of at least 84 %. We hypothesize that the slightly lower value of 76 % for these two categories is caused by the fact that in some cases the annotators interpret the causes and effects with different granularity (e.g., annotators might break some causes and effects down into several sub causes and effects, while some do not). Hence, the annotations differ slightly. The Kappa paradox is particularly evident for the categories Marked and Event Chain. Despite a high agreement of over 90 %, Cohen's Kappa yields a very low value, which "paradoxically" suggests almost no or only fair agreement. A more meaningful assessment is provided by Gwet's AC1 as it did not fail in the case of prevalence and remains close to the percentage of agreement. Across all categories, the mean value is above 0.8, which indicates a nearly perfect agreement. Therefore, we assess our labeled data set as reliable and suitable for further analysis and the implementation of our causality detection approach.

Data Analysis RQ 1-5 are answered by providing descriptive statistics of the distribution of labels for each category. For RQ 6, inferential statistics are applied. Since the hypotheses formulated for each category aim to investigate the independence between the association of a requirement to a specific domain and the distribution in the respective category, a statistical hypothesis test for independence can be used. As both the independent variable (the domain) and the dependent variable (the respective category) are categorical, the Chi-squared test will be used. The category Causality is tested with respect to the full annotated data set. All dependent categories are tested on the causal subset of the data since only causal sentences are annotated in the other categories. For all tests, only domains with at least 100 sentences were selected as eligible strata to confine the hypothesis tests to sufficiently represented domains. This threshold was introduced to RQ 6 to avoid the noise of underrepresented domains. Since the data set was aggregated in RQ 1–5, this change is only necessary for RQ 6. Using the subset of domains as strata implies the degree of freedom of the Chi-squared tests exceeding 2, hence the risk of the multiple comparison problem, i.e., the likelihood for a Type I error in rejecting null hypotheses, arises [2]. For example, when evaluating the null-hypothesis of independence for the dichotomous category Explicit, considering the nine eligible domains with more than 100 causal sentences yields a degree of freedom of 8, as it is calculated as follows [37] (considering that the number of rows is 2 for dichotomous variables):

$$\begin{aligned} dof &= (\text{number of rows} - 1) * (\text{number of columns} - 1) \\ &= (2-1) * (9-1) = 8 \end{aligned} \tag{1}$$

The p-value of the Chi-squared test of this hypothesis is 0.000036, far below the significance level $\alpha = 0.05$, even though the relative number of values in the category Explicit among the eligible domains suggests an equal distribution and therefore independence of the domain, as seen in Fig. 3. Hence, we are not reporting the p-value of the Chi-square hypothesis test, as it is not meaningful in this case. Instead, we apply a Bonferroni correction [2] on the significance level and perform the Chi-squared test in each category for each domain against the sum of all samples outside of the domain, as applied in similar scenarios [33]. Applying the Bonferroni correction to the significance level based on the following formula, [2] yields a significance level that counteracts the large degree of freedom m and reduces the likelihood of Type I errors when refuting null hypotheses:

$$p_c = \frac{\alpha}{m} = \frac{0.5}{8} = 0.00625 \tag{2}$$

The previously calculated p-value for the Chi-square test of independence considering all domains still suggests to reject the null hypothesis. Hence, a post-hoc test similar to [33], where each domain is compared to the sum of all other domains, is applied to reveal, that only the null-hypothesis for the domain *sustainability* can be refuted with a p-value of $0.0001 < 0.00625$, which aligns with Fig. 3. This procedure is applied to all hypotheses of RQ 6.

3.4 Study Results

The results of each labeled category are visualized in Fig. 2. Detailed values for the distribution of labels among all categories and domains are given in Table 11. When interpreting the values, it is important to note that for our analysis we consider complete requirement documents. Consequently, our data set contains records with different contents, which do not necessarily represent all functional requirements. For example, requirement documents also contain non-functional requirements, phrases for content structuring, purpose statements, etc. The results are hence to be interpreted in respect to the content of a full requirements artifact, not only to its functional requirements.

Answer to RQ 1: Figure 2 confirms that causality occurs in requirement documents to a significant extent. About 28 % of the analyzed sentences are causal. From this result, we can conclude that causality is a major linguistic element of requirement artifacts as almost one-third of all sentences are causal.

Answer to RQ 2: The majority (56 %) of causal sentences contained in requirement artifacts represent an *enable* relationship between certain events. Only about 10 % of the causal sentences indicate a *prevent* relationship. *Cause* relationships are found in about 34 % of the annotated data.

Answer to RQ 3: Interestingly, we found that causes and effects occur almost equally often in a *before* and *during* relation. With about 48 %, the *before* relation is the most frequent temporal relation in our data set, but only with a

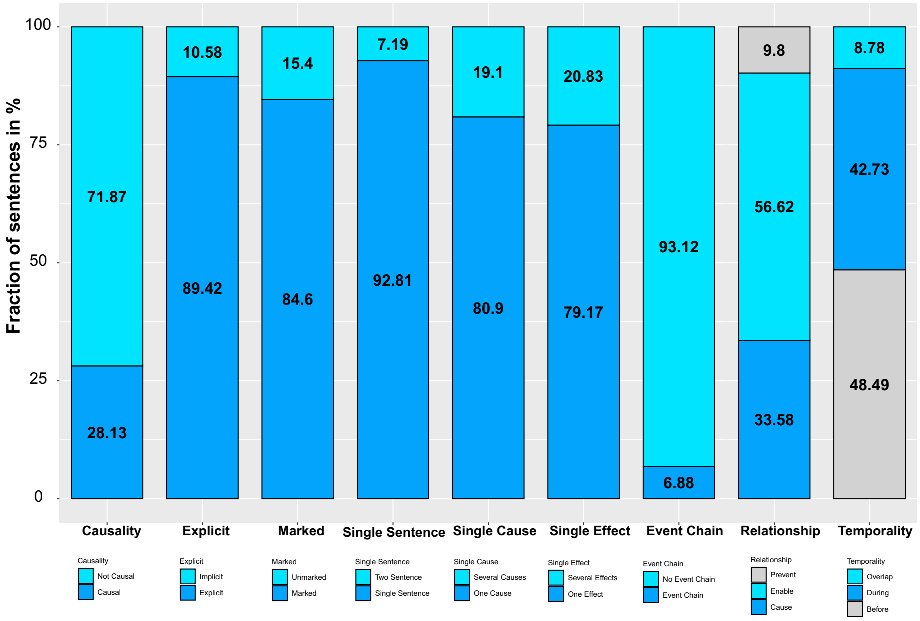


Fig. 2 Annotation results per category. The y axis of the bar plot for the category "Causality" refers to the total number of analyzed sentences. The other bar plots are only related to the causal sentences

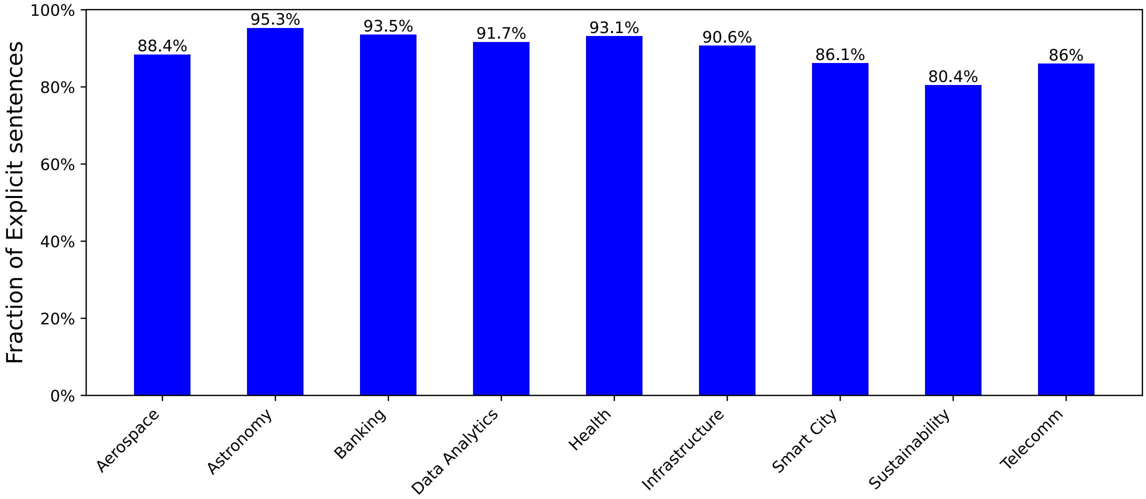


Fig. 3 Percentage of sentences labeled "Explicit" within domains containing at least 100 causal sentences

difference of about 6 % compared to the *during* relation. The *overlap* relation occurred only in a minority (8.78 % of the sentences).

Answer to RQ 4a: The majority of causal sentences contain one or more cue phrases, as Fig. 2 indicates, to indicate the causal relationship between certain events. Only around 15 % of the labeled sentences were categorized as *unmarked* causality.

Answer to RQ 4b: Most causal sentences are *explicit*, i.e., they contain information about both the cause and the effect. Only about 10 % of causal sentences are *implicit*.

Answer to RQ 4c: All causal cue phrases identified in the investigated requirements artifacts are listed in Table 2. The left side of the table shows the cue phrases ordered by word group. On the right side, all verbs used to express causal relations are listed. The verbs are further ordered according to whether they express a *cause*, *enable*, or *prevent* relationship. To assess the ambiguity of a cue phrase x, we formulate a binary classification task: consider all sentences as the sample space. The causal sentences of that sample space represent the relevant elements. The precision of cue phrase x as a selection criterion for causal sentences is the conditional probability, that a sentence from the sample space is causal given that it contains cue phrase x, and hence reflects the ambiguity of the cue phrase:

Table 2 Overview of cue phrases used to indicate causality in requirement documents. **Bold** precision values highlight non-ambiguous phrases that mostly indicated causality (Pr(Causal | X is present in sentence) ≥ 0.8)

Type	Phrase	Causal	Not Causal	Precision
Conjunctions	If	387	41	**0.90**
	As	607	1313	0.32
	Because	78	7	**0.92**
	But	100	204	0.33
	In order to	141	33	**0.81**
	So (that)	88	86	0.51
	Unless	23	4	**0.85**
	While	71	90	0.44
	Once	48	15	0.76
	Except	9	5	0.64
	As long as	12	1	**0.92**
Adverbs	Therefore	61	6	**0.91**
	When	331	64	**0.84**
	Whenever	10	0	**1.00**
	Hence	21	9	0.70
	Where	213	150	0.59
	Then	111	70	0.61
	Since	65	32	0.67
	Consequently	2	6	0.25
	Wherever	5	2	0.71
	Rather	16	30	0.35
	To this/that end	12	0	**1.00**
	Thus	66	17	**0.80**
	For this reason	7	3	0.70
	Due to	91	26	0.78
	Thereby	4	2	0.67
	As a result	11	4	0.73
	For this purpose	1	2	0.33
Pronouns	Which	277	608	0.31
	Who	28	52	0.35
	That	732	1178	0.38
	Whose	16	11	0.59
Adjectives	Only	127	126	0.50
	Prior to	26	20	0.57
	Imperative	1	3	0.25
	Necessary (to)	36	19	0.65
	Given	73	140	0.34
	Following	53	175	0.23
Preposition	For	1209	2753	0.31
	During	327	137	0.70
	After	133	57	0.70
	By	506	1171	0.30
	With	680	1554	0.30
	In the course of	2	1	0.67
	Through	114	204	0.36
	As part of	19	51	0.27
	In this case	18	3	**0.86**
	Before	54	27	0.67
	Until	33	11	0.75
	Upon	25	48	0.34
	In case of	30	7	**0.81**
	In both cases	1	0	**1.00**
	In the event of	15	2	**0.88**
	In response to	6	7	0.46
	In the absence of	8	1	**0.89**
	Within	150	315	0.32
	As far as	4	5	0.44
	According to	21	54	0.28
	Around	25	41	0.37
	From	370	990	0.27
	Based on	56	175	0.24
Cause	Force(s/ed)	21	18	0.54
	Cause(s/ed)	32	10	0.76
	Lead(s) to	5	0	**1.00**
	Reduce(s/ed)	48	28	0.63
	Minimize(s/ed)	28	11	0.72
	Affect(s/ed)	13	19	0.41
	Maximize(s/ed)	11	5	0.69
	Eliminate(s/ed)	8	11	0.42
	Result(s/ed) in	50	43	0.54
	Increase(s/ed)	49	34	0.59
	Decrease(s/ed)	5	8	0.38
	Impact(s)	37	68	0.35
	Degrade(s/ed)	11	2	**0.85**
	Introduce(s/ed)	11	12	0.48
	Enforce(s/ed)	2	1	0.67
	Trigger(s/ed)	11	7	0.61
	Imply	7	14	0.33
	Attain(s/ed)	3	13	0.18
	Create(s/ed)	39	88	0.30
	Impose(s/ed)	7	13	0.35
	Perform(s/ed)	26	60	0.30

Table 2 (continued)

Type	Phrase	Causal	Not Causal	Precision
Enable	Depend(s) on	28	21	0.57
	Require(s/ed)	316	262	0.55
	Allow(s/ed)	187	130	0.59
	Need(s/ed)	98	162	0.38
	Necessitate(s/ed)	7	2	0.78
	Facilitate(s/ed)	29	28	0.51
	Enhance(s/ed)	16	4	**0.80**
	Ensure(s/ed)	145	66	0.69
	Achieve(s/ed)	30	24	0.56
	Support(s/ed)	128	301	0.30
	Enable(s/ed)	75	36	0.68
	Permit(s/ed)	10	13	0.43
	Rely on	3	5	0.38
	Measure(s/ed)	99	247	0.28
	Provide(s/ed)	75	125	0.37
	Get	13	23	0.36
	Meet	42	34	0.55
Prevent	Hinder(s/ed)	1	1	0.50
	Prevent(s/ed)	38	17	0.69
	Avoid(s/ed)	14	23	0.38
	Mitigate(s/ed)	3	8	0.27

$$
\begin{aligned}
&Pr(\text{sentence is causal} \mid \text{sentence contains x}) \\
&\quad = \frac{Pr(\text{sentence is causal} \cap \text{sentence contains x})}{Pr(\text{sentence contains x})}
\end{aligned} \tag{3}
$$

A high precision value indicates a *non-ambiguous* cue phrase, i.e., the occurrence of the cue phrase in a sentence is a strong indicator for the sentence being causal, while low values indicate strongly *ambiguous* cue phrases. Table 2 demonstrates that a number of different cue phrases are used to express causality in requirement documents. Not surprisingly, cue phrases like "if", "because" and "therefore" show precision values of more than 90 %. However, there is a variety of cue phrases that indicate causality in some sentences but also occur in other non-causal contexts. This is especially evident in the case of pronouns. Relative sentences can indicate causality, but not in every case, which is reflected by the low precision value of for example *which*. A similar pattern emerges with regard to the used verbs. Only a few verbs (e.g., "leads to, degrade, and enhance") show a high precision value. Consequently, the majority of used pronouns and verbs do not necessarily indicate a causal relation if they are present in a sentence.

Answer to RQ 5a Fig. 2 illustrates that most causal relations in requirement documents include only a single cause. Multiple causes occur in only 19.1 % of the analyzed causal sentences. The exact number of causes was not documented during the annotation process. However, the participating annotators reported consistently that two to three causes were predominantly prevalent in the case of complex causal relations. More than three causes were rare.

Answer to RQ 5b Interestingly, the distribution of effects is similar to that of causes. Likewise, single effects occur significantly more often than multiple effects. According to the annotators, the number of effects in the case of complex relations is mostly limited to a maximum of two effects. Three or more effects occur rarely.

Answer to RQ 5c Most causal relations can be found in single sentences. Relations, where cause and effect are distributed over two sentences, occur only in about 7 % of the analyzed data. Among the *marked* subset of these sentences ($n = 242$), the cue phrase "therefore" was used most frequently, occurring 58 times. The next-most frequent cue phrase, "thus", appeared only 14 times.

Answer to RQ 5d Fig. 2 shows that *event chains* are rarely used in requirement documents. Most causal sentences contain isolated causal relations, while only roughly 7 % contain *event chains*.

Answer to RQ 6 Figure 4 visualizes the distribution of causality among all domains which are represented with more than 100 sentences. As the percentage of causal sentences ranges from 17.8 % up to 44.4 %, we can assume that causality is indeed a phenomenon occurring in all eligible domains. The Chi-squared test reported in Table 3 suggests rejecting the null hypothesis for domain-independence for 10 out of 14 eligible domains considering the Bonferroni-corrected significance level. We can conclude that causality is a phenomenon observable independent of the domain from which requirements originate, but the extent to which causality occurs differs with statistical significance. For all dependent categories, the domains *Aerospace*, *Astronomy*, *Banking*, *Data Analytics*, *Health*, *Infrastructure*, *Smart City*, *Sustainability*, and *Telecomm* are eligible for consideration as they contain more than 100 causal sentences. On the right side of Table 3, each cell contains the p-value for a Chi-squared test comparing the distribution of the given domain to the rest of the sample. Where the p-value for a given domain and category is lower than the Bonferroni-corrected significance level (denoted for each category as p_c), the cell is prefixed with an asterisk. The Chi-squared test of independence does not suggest to reject the null hypothesis for the categories Single Cause and Event Chain, but the distribution of 4 out of the eligible 9 domains in the category Temporality is significantly different. We can conclude that the distribution of values in all categories is domain-independent to a certain degree: while the complexity of causality is mostly domain-independent, the distribution of temporality differs significantly for a about a third of the eligible domains. A stratified analysis for RQ 4c is reported

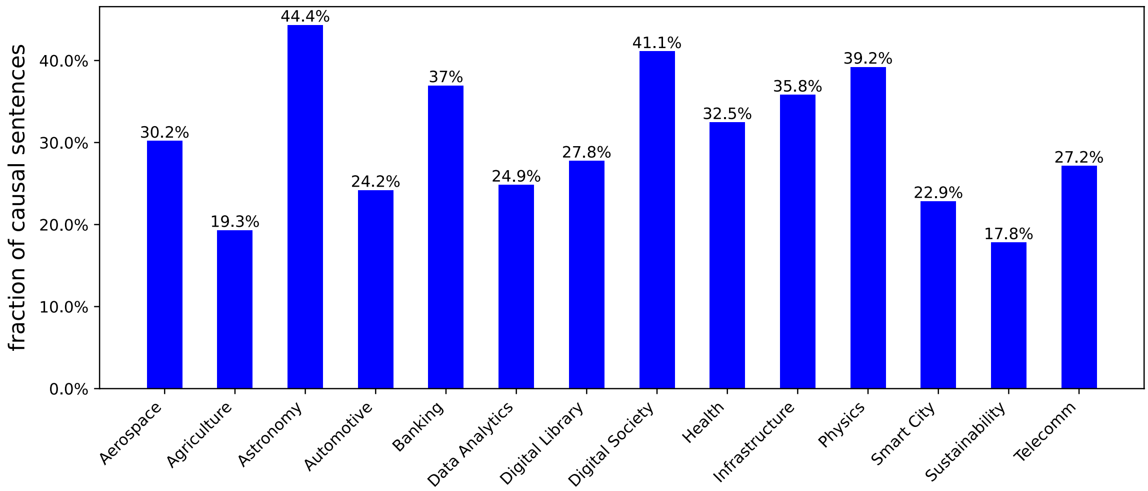


Fig. 4 Distribution of causality among domains

Table 3 Bonferroni-corrected Chi-squared tests of independence from the domain. Cells prefixed with * indicate a category, where the distribution of the given domain differs significantly from the sample

Domain	Causal	Explicit	Marked	Single Cause	Single Effect	Event Chain	Single Sentence	Temporality	Relationship
p_c	3.8E-03	6.3E-03	6.3E-03	6.3E-03	6.3E-03	6.3E-03	6.3E-03	3.1E-03	3.1E-03
Aerospace	*8.0E-05	1.5E-01	8.2E-03	9.8E-02	6.4E-03	7.2E-02	6.3E-01	*3.3E-10	3.1E-02
Agriculture	*7.0E-04	(domain contained less than 100 causal sentences)							
Astronomy	*4.1E-08	6.2E-02	1.0E-02	8.9E-01	2.7E-01	1.1E-01	2.4E-01	*4.4E-05	3.0E-01
Automotive	2.9E-01	(domain contained less than 100 causal sentences)							
Banking	*1.9E-04	1.3E-01	5.3E-01	9.0E-02	2.8E-02	5.3E-01	2.6E-01	*7.8E-06	4.4E-01
Data Analytics	*1.4E-05	3.4E-02	1.3E-02	2.3E-02	*3.7E-03	1.3E-01	3.9E-01	3.6E-01	5.9E-01
Digital Library	9.3E-01	(domain contained less than 100 causal sentences)							
Digital Society	4.4E-03	(domain contained less than 100 causal sentences)							
Health	*9.4E-04	1.4E-02	5.7E-01	7.9E-02	6.8E-01	6.3E-01	7.7E-01	7.0E-02	1.8E-01
Infrastructure	*2.1E-06	5.0E-01	3.2E-01	4.1E-01	1.6E-01	5.0E-01	6.8E-01	6.1E-01	*6.7E-05
Physics	*2.1E-04	(domain contained less than 100 causal sentences)							
Smart City	*8.4E-07	5.9E-02	*2.0E-05	1.3E-02	3.5E-01	3.2E-01	*2.3E-03	*3.1E-05	3.9E-01
Sustainability	*1.4E-14	*1.2E-04	*2.3E-04	8.7E-01	5.4E-01	1.4E-02	*5.7E-03	3.5E-01	1.9E-01
Telecomm	5.7E-01	2.2E-01	7.3E-01	5.2E-01	3.1E-01	1.8E-02	3.5E-02	1.0E-01	7.1E-01

in Table 4a and shows considerable differences in the usage of cue phrases in the domains, but also a degree of overlap: the cue phrase *if* is among the five most frequent cue phrases in all domains, closely followed by the cue phrases *when* and *where*. The stratified frequencies align with the overall distribution reported in Table 2 and lead to the assumption, that the distribution of cue phrases is mostly domain independent. When looking at the most precise cue phrases per domain in Table 4b and the least precise cue phrases per domain in Table 4c, the cue phrases also reflect the findings from the overall distribution: precise cue phrases like *if*, *when*, and *because* as well as infrequent, but precise causative verbs are equally represented in the domains just as imprecise cue phrases like *for* or *by*. We conclude that despite slight domain-specific variations, the results for RQ 4c are also domain-independent[2].

3.5 Implications for causality detection and extraction

Based on the results of our case study, we draw the following conclusions: Causality is prevalent in requirements artifacts and therefore matters in requirements engineering, which motivates the necessity of not only an effective and reliable approach for the automatic detection and extraction of

[2] More extensive tables reporting on the frequency and precision of cue phrases in eligible domains are included in the replication package.

Table 4 Distribution and precision of cue phrases in eligible domains

(a) Relative frequency of cue phrases within one domain

Domain	most frequent	2nd most frequent	3rd most frequent	4th most frequent	5th most frequent
Aerospace	during (8.5%)	when (8.4%)	if (7.5%)	in order to (3.7%)	after (3.0%)
Astronomy	for (10.2%)	during (9.1%)	allow (9.1%)	in (5.7%)	to (5.7%)
Banking	if (11.2%)	to ensure (8.8%)	once (7.2%)	allow (5.6%)	through (4.8%)
Data Analytics	if (10.7%)	when (9.4%)	where (7.9%)	for (5.2%)	during (4.2%)
Health	when (11.5%)	if (11.5%)	during (10.5%)	for (4.6%)	after (3.8%)
Infrastructure	where (16.5%)	if (15.1%)	to ensure (5.6%)	for (5.3%)	then (4.9%)
Smart City	in order to (10.7%)	when (7.4%)	if (7.1%)	therefore (4.9%)	that (4.4%)
Sustainability	therefore (8.6%)	in order to (7.0%)	if (5.9%)	for (5.4%)	where (4.3%)
Telecomm	if (8.7%)	during (8.0%)	in order to (6.0%)	when (5.3%)	in case of (4.7%)

(b) Most precise cue phrases of each eligible domain

Domain	most precise	2nd most precise	3rd most precise	4th most precise
Aerospace	imposes (100.0%)	as far as (100.0%)	result from (100.0%)	in this case (100.0%)
Agriculture	since (100.0%)	whose (100.0%)	before (100.0%)	when (100.0%)
Astronomy	during (100.0%)	in the event of (100.0%)	so that (100.0%)	attain (100.0%)
Automotive	when (100.0%)	in order to (100.0%)	therefore (100.0%)	whose (100.0%)
Banking	in order to (100.0%)	reduce (100.0%)	will be required (100.0%)	since (100.0%)
Data Analytics	as long as (100.0%)	increases (100.0%)	unless (100.0%)	so that (100.0%)
Digital Library	only for (100.0%)	cause (100.0%)	because (100.0%)	unless (100.0%)
Digital Society	in order to (100.0%)	if (100.0%)	due to (100.0%)	allows (100.0%)
Health	allows (100.0%)	so that (100.0%)	in the event of (100.0%)	as a result (100.0%)
Infrastructure	lead to (100.0%)	prevent (100.0%)	whose (100.0%)	until (100.0%)
Physics	is needed (100.0%)	after (100.0%)	therefore (100.0%)	when (100.0%)
Smart City	result in (100.0%)	to measure (100.0%)	increase (100.0%)	thereby (100.0%)
Sustainability	will require (100.0%)	enables (100.0%)	ensures (100.0%)	because (100.0%)
Telecomm	allows (100.0%)	enables (100.0%)	during (100.0%)	lead to (100.0%)

(c) Least precise cue phrases of each eligible domain

Domain	least precise	2nd least precise	3rd least precise	4th least precise
Aerospace	according to (14.3%)	to get (20.0%)	to mitigate (20.0%)	based on (24.1%)
Agriculture	on (26.1%)	that (27.1%)	at (28.2%)	allows (33.3%)
Astronomy	from (30.0%)	by (31.0%)	within (40.0%)	where (50.0%)
Automotive	which (10.0%)	for (20.0%)	on (24.2%)	that (31.8%)
Banking	which (21.4%)	to provide (25.0%)	create (28.6%)	by (28.8%)
Data Analytics	to meet (16.7%)	imply (20.0%)	following (20.4%)	but (20.5%)
Digital Library	allow (11.1%)	for (27.6%)	that (30.3%)	with (31.8%)
Digital Society	for (45.8%)	for this reason (50.0%)	where (62.5%)	allow (100.0%)
Health	in this (16.7%)	around (20.0%)	based on (22.2%)	through (25.0%)
Infrastructure	following (18.2%)	to provide (25.0%)	on (41.3%)	in (42.8%)
Physics	while (33.3%)	from (34.8%)	in this (42.9%)	for (43.6%)
Smart City	within (11.1%)	to perform (15.4%)	who (15.4%)	where (15.8%)
Sustainability	within (3.8%)	with (13.7%)	to provide (14.3%)	while (14.3%)
Telecomm	to provide (16.7%)	which (19.4%)	so that (20.0%)	given (20.0%)

causal requirements, but also an investigation of the impact causality in requirements artifacts has. The complexity of causal relations is confined since they usually consist of a single cause and effect relationship in all observed, eligible domains. However, for an approach that aims to extract the causal relationship to be applicable in practice, it needs to comprehend also more complex relations containing at least two to three and at best an arbitrary number of causes and effects. Understanding conjunctions, disjunctions, and negations is consequently imperative to fully capture the relationships between causes and effects and ensure the applicability of a detection and extraction approach. *Two-sentence causality* and *event chains* occur only rarely. Thus, both aspects can initially be neglected in the development of the approaches and preserve coverage of more than 92 % of the analyzed sentences. The dominance of *explicit* over *implicit* causal relations in the observed sentences simplifies the detection and extraction of causality. The information about both causes and effects is embedded directly in the sentences so that an approach requires little or no *implicit* knowledge. The analysis of the precision values reveals that most of the used cue phrases are ambiguous. Consequently, automatic detection and extraction methods require a deep understanding of language as the presence of certain cue phrases is insufficient as an indicator for causality. Instead, a combination of the syntax and semantics of the sentence has to be considered to reliably detect causal relations.

3.6 Threats to validity

Internal Validity A threat to the internal validity is the annotation process itself as any annotation task is subjective to a certain degree. This is especially relevant for more ambiguous categories like Explicit, as *implicit* causality is difficult to determine. Two mitigation strategies were performed to minimize the bias of the annotators: First, we conducted a workshop prior to the annotation process to ensure a common understanding of causality. Second, we assessed the inter-rater agreement by using multiple metrics (Cohen's Kappa, Agreement Score, and Gwet's AC1). However, it has to be noted that all categories except Causality are dependent on a sentence's classification regarding that category, which may imply a confounding factor for the inter-rater agreement on the other categories. This manifests in the calculation of the inter-rater agreement, where all categories except Causality are calculated based on the 499 causal sentences. We argue, however, that the other categories are irrelevant for non-causal sentences as they only refer to the causal relation contained by a sentence, hence this confounding factor is deemed minimal. Apart from that, the inter-rater agreement is not domain-specific, which implies that it is not possible to identify, whether certain domains caused more disagreement among the raters. We deem the general inter-rater agreement reported in Table 1 sufficient but recommend considering this aspect for replications and future studies intensifying the domain-dependent aspect of causality. Furthermore, restricting the manual detection of causal relations to a span of a maximum of two sentences poses also a threat to internal validity, as the potential existence of causal relations that are spread across more than two sentences can neither be confirmed nor denied based on our investigation. We see this threat to be minimal as the relationship between one-sentence-causality and two-sentence-causality allows for the assumption, that the further elements of a causal relation are spread apart, the more unlikely the existence of such a causal relation is. Extrapolating from the low number of sentences categorized as two-sentence-causality gives us reason to assume that disregarding causal relations spread across three sentences or more is negligible for this initial case study.

External Validity To achieve reasonable generalizability, we selected requirements documents from different domains and years. As Fig. 1 shows, our data set covers a variety of domains, but the distribution of the sentences is imbalanced. The domains aerospace, data analytics, and smart city account for a large share in the data set (9,724 sentences), while the other 15 domains are rather underrepresented. We mitigate this threat to validity by including a domain-specific investigation reported in the scope of RQ 6, which confirms that the occurrence of causality is to a large degree domain-independent. Future studies should, however, expand to more documents emerging from underrepresented domains to allow a more general reflection upon different aspects of causality in requirements documents.

4 Approach: detecting causal requirements

This section presents the implementation of our causal classifier. To this end, we describe a variety of applied methods followed by a report of the results of our experiments, in which we compare the performance of the individual methods and draw a conclusion in regards to applicability.

4.1 Methods

Rule-based Approach Instead of using a random classifier as the baseline approach, we involve simple regex expressions for causality detection. We iterate through all sentences in the test set and check if one of the phrases listed in Table 2 is contained. In the positive case, the sentence is classified as causal and vice versa. As discussed in Sect. 2, the classification of a sentence as causal based on the occurrence of a cue phrase—which the baseline approach represents—is reasonable to assume.

Fig. 5 Input sequences used for our different BERT fine-tuning models. POS tags are marked blue, and DEP tags are marked orange

BERT$_{\text{Base}}$: If the process fails, an error message is shown.
BERT$_{\text{POS}}$: If_**SCONJ** the_**DET** process_**NOUN** fails_**VERB** ,_**PUNCT** an_**DET** error_**NOUN** message_**NOUN** is_**AUX** shown_**VERB** ._**PUNCT**
BERT$_{\text{DEP}}$: If_**mark** the_**det** process_**nsubj** fails_**advcl** ,_**punct** an_**det** error_**compound** message_**nsubjpass** is_**auxpass** shown_**ROOT** ._**punct**

Machine Learning-based Approach As a second approach, we investigate the use of *supervised* ML models that learn to predict causality based on the labeled data set. Specifically, we employ established binary classification algorithms: Naive Bayes (NB), Support Vector Machines (SVM), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Ada Boost (AB), and K-Nearest Neighbor (KNN). To determine the best hyperparameters for each binary classifier, we apply Grid Search, which fits the model on every possible combination of hyperparameters and selects the most performant. We use two different methods as word embeddings: Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). In Table 5, we report the classification results of each algorithm as well as the best combination of hyperparameters.

Deep Learning-based Approach With the rise of Deep Learning (DL), more and more researchers are using DL models for Natural Language Processing (NLP) tasks. In this context, the Bidirectional Encoder Representations from Transformers (BERT) model [8] is prominent and has already been used for question answering and named entity recognition. BERT is pre-trained on large corpora and can therefore easily be fine-tuned for any downstream task without the need for much training data (Transfer Learning). In our paper, we make use of the fine-tuning mechanism of BERT and investigate to which extent it can be used for causality detection of requirement sentences. First, we tokenize each sentence. BERT requires input sequences with a fixed length (maximum 512 tokens). Therefore, for sentences that are shorter than this fixed length, padding tokens (PAD) are inserted to adjust all sentences to the same length. Other tokens, such as the classification (CLS) token, are also inserted in order to provide further information of the sentence to the model. CLS is the first token in the sequence and represents the whole sentence (i.e., it is the pooled output of all tokens of a sentence). For our classification task, we mainly use this token because it stores the information of the whole sentence. We feed the pooled information into a single-layer feedforward neural network that uses a softmax layer, which calculates the probability that a sentence is causal or not. We tune BERT in three different ways and investigate their performance:

- **BERT$_{\text{Base}}$** In the base variant, the sentences are tokenized as described above and put into the classifier. To choose a suitable fixed length for our input sequences, we analyzed the lengths of the sentences in our data set. Even with a fixed length of 128 tokens, we cover more than 97 % of the sentences. Sentences containing more tokens are shortened accordingly. Since this is only a small amount, only a little information is lost. Thus, we chose a fixed length of 128 tokens instead of the maximum possible 512 tokens to keep BERT's computational requirements to a minimum.
- **BERT$_{\text{POS}}$** Studies have shown that the performance of NLP models can be improved by providing explicit prior knowledge of syntactic information to the model [48]. Therefore, we enrich the input sequence with syntactic information and feed it into BERT. More specifically, we add the corresponding part-of-speech (POS) tag to each token by using the spaCy NLP library [27]. One way to encode the input sequence with the corresponding POS tags is to concatenate each token embedding with a hot encoded vector representing the POS tag. Since the BERT token embeddings are high-dimensional, the impact of a single added feature (i.e., the POS tag) would be low. Contrary, we hypothesize that the syntactic information has a higher impact if we annotate the input sentences directly with the POS tags and then put the annotated sentences into BERT. This way of creating linguistically enriched input sequences has already proven to be promising during the development of the NLPL word embeddings [10]. Fig. 5 shows how we incorporated the POS tags into the input sequence. By extending the input sequence, the fixed length for the BERT model has to be adapted accordingly. After further analysis, a length of 384 tokens proved to be reasonable.
- **BERT$_{\text{DEP}}$** Similar to the previous fine-tuning approach, we follow the idea of enriching the input sequence by linguistic features. Instead of using the POS tags, we use the dependency (DEP) tags (see Fig. 5) of each token. Thus, we provide knowledge about the grammatical structure of the sentence to the classifier. We hypothesize that this knowledge has a positive effect on the model performance, as a causal relation is a specific grammatical structure (e.g., it often contains an adverbial clause) and the classifier can learn causal specific patterns in the grammatical structure of the training instances. The fixed token length was also increased to 384 tokens.

Fig. 6 Implementation and Evaluation Procedure of our Binary Classifier

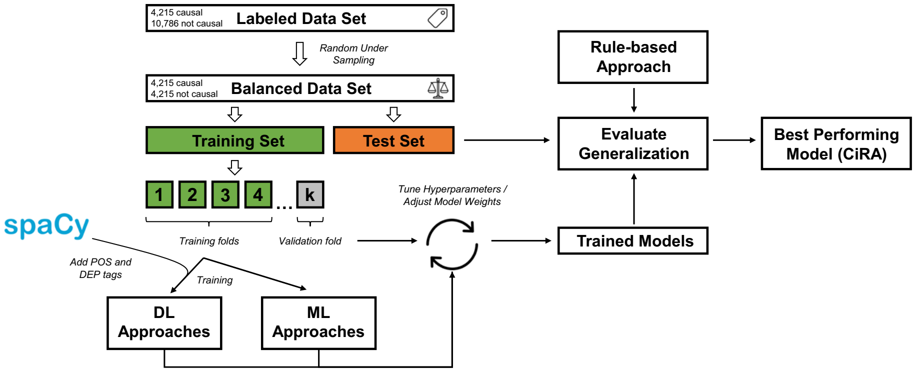


Table 5 Recall, Precision, F_1 scores (per class) and Accuracy. We report the averaged scores over five repetitions. Best results for each metric are highlighted in **bold**

Rule based			Causal (Support: 435)			Not Causal (Support: 408)			
		Best hyperparameters	Recall	Precision	F1	Recall	Precision	F1	Accuracy
			0.65	0.66	0.66	0.65	0.63	0.64	0.65
ML based	NB	Alpha: 1, fit_prior: True, embed: BoW	0.71	0.7	0.71	0.68	0.69	0.69	0.7
	SVM	C: 50, gamma: 0.001, kernel: rbf, embed: BoW	0.68	0.8	0.73	0.82	0.71	0.76	0.75
	RF	Criterion: entropy, max_features: auto, n_estimators: 500, embed: BoW	0.72	**0.82**	0.77	**0.84**	0.74	0.79	0.78
	DT	Criterion: gini, max_features: auto, splitter: random, embed: TF-IDF	0.65	0.68	0.66	0.67	0.65	0.66	0.66
	LR	C: 1, solver: liblinear, embed: TF-IDF	0.71	0.78	0.74	0.79	0.72	0.75	0.75
	AB	Algorithm: SAMME.R, n_estimators: 200, embed: BoW	0.67	0.78	0.72	0.8	0.7	0.75	0.74
	KNN	Algorithm: ball_tree, n_neighbors: 20, weights: distance, embed: TF-IDF	0.61	0.68	0.64	0.7	0.63	0.66	0.65
DL based	$\text{BERT}_{\text{Base}}$	batch_size: 16, learning_rate: 2e-05, weight_decay: 0.01, optimizer: AdamW	0.83	0.80	0.82	0.78	0.82	0.80	0.81
	BERT_{POS}		0.82	0.76	0.79	0.71	0.83	0.77	0.78
	BERT_{DEP} (CiRA)		**0.85**	0.81	**0.83**	0.79	**0.84**	**0.81**	**0.82**

4.2 Evaluation procedure

Our labeled data set is imbalanced as only 28.1 % are positive samples. To avoid the class imbalance problem, we apply Random Under Sampling (see Fig. 6). We randomly select sentences from the majority class and exclude them from the data set until a balanced distribution is achieved. Our final data set consists of 8,430 sentences of which 4,215 are causal and the other 4,215 are non-causal. We follow the idea of cross-validation and divide the data set into a training, validation, and test set. The training set is used for fitting the algorithm, while the validation set is used to tune its parameters. The test set is utilized for the evaluation of the algorithm based on real-world unseen data. We opt for 10-fold cross-validation as a number of studies have shown that a model that has been trained this way demonstrates low bias and variance [29]. We use standard metrics for evaluating our approaches: Accuracy, Precision, Recall, and F_1 score [29]. Since a single run of a *k*-fold cross-validation may result in a noisy estimate of model performance, we repeat the cross-validation procedure five times and average the scores from all repetitions (see Tab. 5). When interpreting the metrics, it is important to consider which misclassification (False Negative or False Positive) matters most, respectively, causes the highest costs. Since causality detection is supposed to be the first step toward automatic causality extraction, we favor Recall over Precision. A high Recall corresponds to a greater degree of automation of causality extraction because it is easier for users to discard False Positives than to manually detect False Negatives. Consequently, we seek high Recall to minimize the risk of missed causal sentences and acceptable Precision to ensure that users are not overwhelmed by False Positives.

4.3 Experimental results

Table 5 demonstrates the inability of the rule-based baseline approach to distinguish between causal (F_1 score: 66%) and non-casual (F_1 score: 64%) sentences. This coincides with our observation from the first case study that classifying sentences as causal or non-causal based on the occurrence of cue phrases is not suitable for causality detection. In comparison, most ML-based approaches (except KNN and DT) show a better performance. The best performance in this category is achieved by RF with an Accuracy of 78% (gain of 13 % compared to baseline approach). The overall best classification results are achieved by our DL-based approaches. All three variants were trained with the hyperparameters recommended by Devlin et al. [8]. Even the vanilla $\mathbf{BERT_{Base}}$ model shows a great performance in both classes (F_1 score ≥ 80 % for causal and non-causal). Interestingly, enriching the input sequences with syntactic information did not result in a significant performance boost. $\mathbf{BERT_{POS}}$ even has a slightly worse Accuracy value of 78% (difference of 2% compared to $\mathbf{BERT_{Base}}$). An improvement of the performance can be observed in the case of $\mathbf{BERT_{DEP}}$, which has the best F_1 score for both classes among all the other approaches and also achieves the highest Accuracy value of 82%. Compared to the rule-based and ML-based approaches, $\mathbf{BERT_{DEP}}$ yields an average gain of 11.06% in macro-Recall and 11.43% in macro-Precision. Interesting is a comparison with $\mathbf{BERT_{Base}}$. $\mathbf{BERT_{DEP}}$ shows better values across all metrics, but the difference is only marginal. This indicates that $\mathbf{BERT_{Base}}$ already has a deep language understanding due to its extensive pre-training and therefore can be tuned well for causality detection without much further input. However, over all five runs, the use of the dependency tags shows a small but not negligible performance gain—especially regarding our main decision criterion: the Recall value (85% for causal and 79% for non-causal). Therefore, we choose $\mathbf{BERT_{DEP}}$ as our final approach (CiRA).

5 Case study 2: effects of causality

After discussing first empirical evidence on the extent and complexity to which causality is used in NL requirements in our first case study (C1) and constructing a reasonably effective approach for automatic causality detection (C2), we aim to corroborate the relevance of causality for requirements in a second case study (C3). Here, we investigate the impact of causal relations on the features of requirements, where we consider features to be observable attributes of individual requirements (e.g., their lead-time). This investigation emerges from an ongoing academia-industry collaboration in a larger context. Our exploratory case study has two goals: first, we aim to demonstrate an independent use case of the automatic causality detection approach. While automatic causality detection as presented in Sect. 4 can be used as a precursor to automatic causality extraction and therefore as one step in a pipeline toward automatic test case generation, we explore considering the occurrence of causality as an aspect of requirements quality and consequently automatic causality detection as a metric to estimate requirements quality. An effective tool-supported approach for detecting causality in NL requirements allows exploring the eligibility of causality as an aspect of requirements quality. Gathering the first empirical evidence toward this is the second goal of this exploratory case study. First empirical evidence can be gathered by investigating the correlation between the occurrence of causality in requirements and features of these requirements.

5.1 Research questions

We are interested in the impact that the occurrence of causal relations in natural language requirements has on important features of these requirements. Empirical evidence for an impact of causality on these features would allow the assumption that the use of causality in a NL requirement contributes to the requirement's quality. While a definite connection cannot be determined based on a correlation analysis alone, this exploratory case study rather aims toward providing first insights into the feasibility of using causality as a quality aspect for requirements and opening up a more detailed discussion regarding specific features. We select the following features for our analysis:

- *Lead-Time*: the time from the inception until the completion of a requirement.
- *Consolidated state*: the type of final state in which the requirement ends in.
- *Volatility*: number of state changes which the requirement undergoes.

The selection of attributes is inspired by research with comparable objectives, which used lead-time and the resulting consolidated state [41] as well as the volatility [51] of requirements as dependent variables to estimate the impact of requirements attributes on the downstream development process. A data set eligible for this evaluation needs to provide information in form of a state log, where each entry in the log documents the author, timestamp, and state code. The state codes represent the different states which a requirement traverses during its life cycle from its inception to its completion. The lead-time consequently constitutes the time delta between the first and the last state log entry.

The consolidated state is the final state of the state log. The volatility denotes the number of entries in the state log, as it directly correlates with the number of additional development cycles the requirement has to traverse (e.g., by being pushed back to earlier states when repeating one development cycle). We want to investigate whether a statistically significant difference can be determined in the distribution of lead-time, consolidated state, and volatility between requirements that use causal relations versus requirements that do not. To this extent, we aim at providing answers to the following research questions (RQ):

- *RQ 7*: Does the use of causality in an NL requirement correlate with its lead-time?
- *RQ 8*: Does the use of causality in an NL requirement correlate with its consolidated state?
- *RQ 9*: Does the use of causality in an NL requirement correlate with its volatility?

These attributes have been chosen as an eligible representation of the requirements' comprehensibility and degree of ambiguity. As elaborated in earlier sections, we hypothesize that the clear semantic structure of a causal relation promotes comprehensibility and mitigates the ambiguity of requirements. This would result in shorter lead times, a greater likelihood of a successful outcome, and less volatility, as the requirement has to undergo fewer iterations in the development life-cycle.

5.2 Study design

Study Objects The study is performed on a data set for an industrial, proprietary project. The owning, multi-national case company develops and globally distributes software-intensive products for a B2C market. The number of engineers involved with the product line of the data set in question varied from 1000 to 4000 worldwide. The original data set, pre-processed by Olsson et al. [41], contains 4446 requirements collected in 2016[3]. The data set has been chosen because it contains the aforementioned features necessary for the evaluation, other than the data sets used for the first case study: most of the requirements in the data set contain a state log documenting the requirement's life cycle from its inception as a *New Feature* (NF) to its final state as either *Execution completed* (EC) or *Discarded* (D). The newly created requirements undergo the initial triage in a state called M0. Next, upon considered viable and sufficiently justified, the requirement candidates are prioritized in a project prioritization state (similar to backlog prioritization), called M1. Finally, the prioritized requirements are hand-shaken with the developer teams in a state called M2 [18]. When a requirement is unclear at the M2 state, it is pushed back to M1 for re-prioritization. Similarly, a requirement is pushed back to M0 when questions and uncertainties arise during requirements prioritization. These backward transitions unusually increase the lead-time. The features relevant for this study are described in Table 6. Further attributes were generated based on the existing features:

Table 6 Features of the data set

Feature	Description	Datatype
ID	Unique identifier of the requirement	Numeric
Description	Textual description containing a varying amount of NL sentences	Text
State log	History of state changes	Categorical list
Date of creation	Inception date of the requirement	Date

- *Sentences*: The number of sentences occurring in the *Description* field is counted via sentence tokenization.
- *Causal Relations*: The number of causal sentences in the *Description* field is counted by applying the CiRA-tool presented in Sect. 4 to each sentence.
- *Lead-Time*: The lead-time is calculated as the time frame between the first and the last entry of the *state log*.
- *Volatility*: Number of decisions counted as the number of entries in the *state log*

The data set of 4446 requirements was further pre-processed to serve the application in this second case study. Three additional filters have been applied: (1) requirements, for which the state log did not exist, were discarded. (2) Requirements with a state log authored by exactly one, specific individual, were discarded. The entries to these state logs were due to database migrations and do not contain actual information on the requirements life cycle. (3) Requirements with only one entry in the state log were discarded. These requirements do not allow calculating a meaningful lead-time. In total, 815 requirements were discarded due to the pre-processing, leaving 3631 requirements for the analysis. Table 7 lists further details on the filtering process.

Data Analysis The research questions can be translated into statistically verifiable (i.e., refutable) hypotheses. We therefore formulate the following null hypotheses:

- $\mathbf{H_{1_0}}$: Requirements containing different amounts of causality have the same distribution of lead-time.

[3] The proprietary data set cannot be disclosed at the time of submission as it contains critical company information.

Table 7 Pre-processing steps

ID	Filter	Removed	Remaining
1	Missing consolidated state log	176	4270
2	Specific, invalid author	185	4085
3	Singular state log entry	454	3631
	Total	**815**	**3631**

Table 8 Distribution of sentences according to granularity G1 and G2

	n_{causal}	n_{req}
Non-causal	[0]	1000
Causal	[1, 3]	2059
	[4, 6]	457
	[7, 9]	96
	[10, 12]	15
	[13, 15]	2
	[16, 18]	1
	[19, 21]	1
Total	[0, 21]	3631

Table 9 Distribution of sentences according to granularity G1 and G3

$n_{sentences}$	[1, 3]	[4, 7]	[8, max]	[1,max]
Causal	542	933	1156	2631
Non-causal	514	331	175	1000
Total	1056	1244	1331	3631

- $\mathbf{H_{2_0}}$: Requirements containing different amounts of causality have the same distribution of consolidated states.
- $\mathbf{H_{3_0}}$: Requirements containing different amounts of causality have the same distribution of volatility.

In all hypotheses, the input variable *contained amounts of causality* is tested on two levels of granularity: (G1) binary (containing at least one causal sentence vs. containing no causal sentence) and (G2) in batches (ranges of number of causal sentences). Furthermore, granularity G1 is extended to the third level of granularity (G3) where the data set is split into three subsets containing requirements of different sentence sizes. The distribution of causal sentences according to the different levels of granularity is given in Tables 8 and 9. Where the binary granularity G1 serves to investigate the general effect of causality and the batch granularity G2 refines this relation, the extended granularity G3 normalizes the effect according to requirement size. All hypotheses are reported using descriptive statistics and evaluated using inferential statistics. The hypothesis of independence is calculated using the Mann–Whitney U test on the binary level of granularity (G1 and G3) for the interval scale variable of lead-time and volatility, and using the Chi-square test for the categorical variable of consolidated state. For batch granularity G2, the Kruskal–Wallis test is used [19]. All statistical tests of independence are evaluated with a significance level $\alpha = 0.05$. Where a statistical tests suggest to reject the null hypothesis of independence, the effect size of the correlation is quantified using Cohen's d for binary granularities G1 and G3 [47] and Eta-squared for batch granularity G2 [7]. These measures allow categorizing the magnitude of the correlation effect.

5.3 Study results

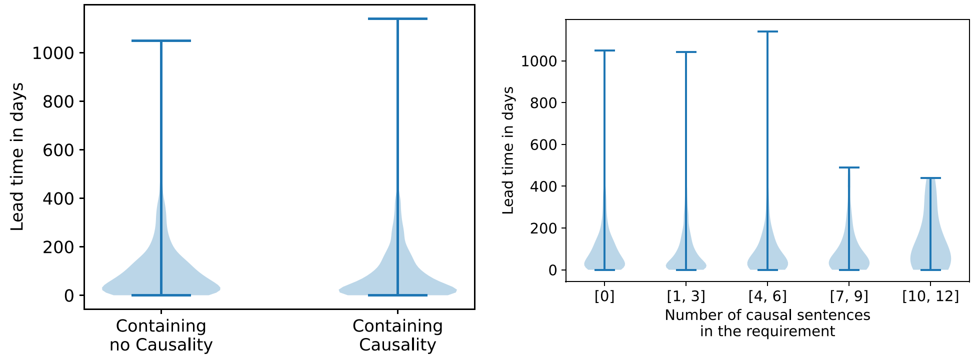
All study results are reported in Table 10 and explained in further detail in the following Sections.

Correlation between causality and lead-time Fig. 7a displays the distribution of lead-time in the two binary groups in the form of violin plots and indicates that the lead-time of requirements containing at least one causal sentence is on average lower than the lead-time of requirements without any causality. The Mann–Whitney U test of independence yields a p-value of 0.00038 far below the significance level $\alpha = 0.05$, rejecting the null hypothesis of similar distribution and corroborating the found difference. Cohen's d quantifying the effect size yields 0.0514, which categorizes the effect size of the correlation as *small*. For granularity G2, only batches containing more than 10 sentences were included, which leads to discarding all batches containing more than 12 causal sentences due to statistical insignificance. The results of evaluating H_{1_0} on a finer level of granularity G2 shows that the increased usage of causal sentences has a positive effect on reducing the average lead-time up until the point of using 10 or more causal sentences as shown in Fig. 7b. The Kruskal–Wallis test of independence suggests to reject the null hypothesis of similar distribution with a p-value of 0.0082, but the Eta-squared value of 0.0010 categorizes the correlation as *negligible* [7]. Evaluating the difference in distribution on granularity G3 reveals that the size of the requirement is a contributing factor to the correlation between the occurrence of causality and the lead-time: while the use of causality in small requirements has a negative effect on the lead-time, the opposite is observable for medium and large requirements, as illustrated in Fig. 8. The null hypothesis of independence is accepted for small requirements with $p = 0.34$ and rejected for medium and large requirements with $p = 0.0008$ and $p = 0.008$, respectively. The effect sizes are 0.12 and 0.15 based on the Cohen's d measure.

Correlation between causality and consolidated state The data set uses 20 categories for the variable *consolidated state*, of which most represent intermediate states. The data set of 3442 requirements is filtered for all requirements in final states, which are *execution completed* (EC)

Table 10 P- and Cohen's d value for evaluating the respective null-hypothesis with the given granularity. Cells prefixed with * indicate where the null-hypothesis has been rejected (given significance level $\alpha = 0.05$)

Hypothesis	Measure	G1	G2	G3		
				[1, 3]	[4, 7]	[8, max]
$\mathbf{H_{1_0}}$	*p* value	*0.0004	*0.0082	0.3434	*0.0008	*0.0084
	Effect size	0.0514	0.0010	–	0.1181	0.1496
$\mathbf{H_{2_0}}$	*p* value	0.8497	0.0668	0.1368	0.1456	0.7661
	Effect size	–	–	–	–	–
$\mathbf{H_{3_0}}$	*p* value	*0.0105	0.0856	0.2184	*0.0011	0.1261
	Effect size	0.0742	–	–	0.1843	–



(a) Binary granularity G1

(b) Batch granularity G2

Fig. 7 Distribution of lead-time (H_{1_0})

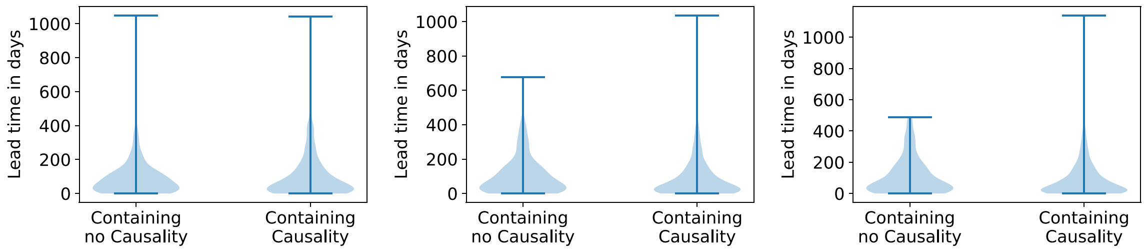
and *discarded* (D), respectively, positive and negative final state. Only the 1157 requirements in these two final states (EC: 591, D: 566) were considered for this evaluation. Fig. 9a illustrates the distribution of consolidated states on binary granularity G1. The Chi-square test of independence yields a p-value of 0.85 and does therefore not allow to reject the null hypothesis. Increasing the granularity to batches as displayed in Fig. 9b suggests a positive trend in the correlation between the occurrence of causality and a successful consolidated state, but the Kruskal–Wallis test does not allow to reject the null hypothesis of similar distribution with a p-value of 0.067. At granularity G3, the consolidated states of requirements of different sizes correlate negatively with the occurrence of causality for small and medium requirements with only a slight positive correlation for large requirements, as seen in Fig. 10. The null hypothesis of independence can, however, not be rejected with p-values of 0.14, 0.15, and 0.77.

Correlation between causality and volatility Fig. 11a displays the distribution of the volatility metric in the two binary groups as violin plots. Overall, a slight correlation between the occurrence of causality and the volatility of a requirement can be observed, which is corroborated by the rejected test of independence with a p-value of 0.01 and an effect size of 0.07. Investigating this effect at batch granularity G2 in Fig. 11b, however, reveals that this positive correlation is constrained to requirements with a low occurrence of causal sentences, where requirements with many causal sentences show a trade-off of higher average volatility despite a smaller overall range of volatility values. The Kruskal–Wallis test of independence does not allow to reject the null-hypothesis of similar distribution with a p-value of 0.086. Investigating the correlation at granularity G3 as displayed in Fig. 12, this trade-off is again visible and emphasizes the positive correlation between the occurrence of causality and the volatility of requirements for medium-sized requirements. This is confirmed by the independence tests, where the null hypothesis can be rejected for medium requirements with a p-value of 0.001 and effect size of 0.18, but not for small or large requirements with p-values of 0.22 and 0.13, respectively.

5.4 Implications

Impact of causality The results of the second case study show an already existing positive correlation between the occurrence of causal relations and the features of requirements artifacts. These results motivate further, in-depth investigations corroborating the relationship between the occurrence of causality and features of these requirements, suggesting the feasibility of considering causality as an aspect of requirements quality. Both the direct insights and the consequent hypothesis for future research are discussed in more detail in the following paragraphs.

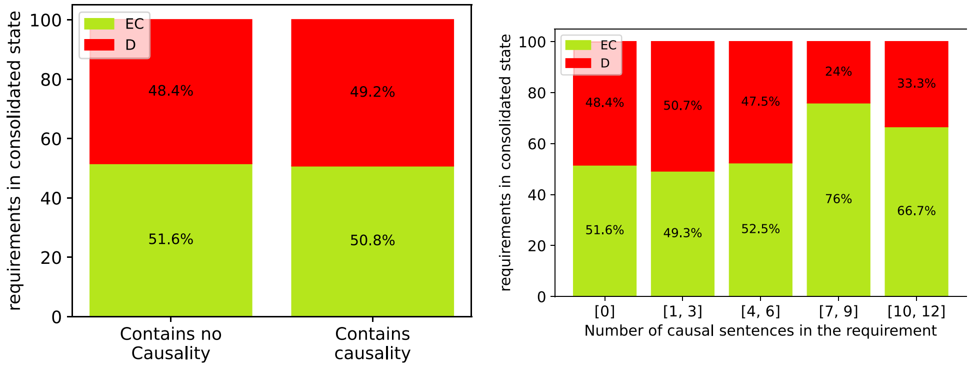
Answer to RQ 7: The use of causality correlates slightly with smaller lead-times of requirements and therefore



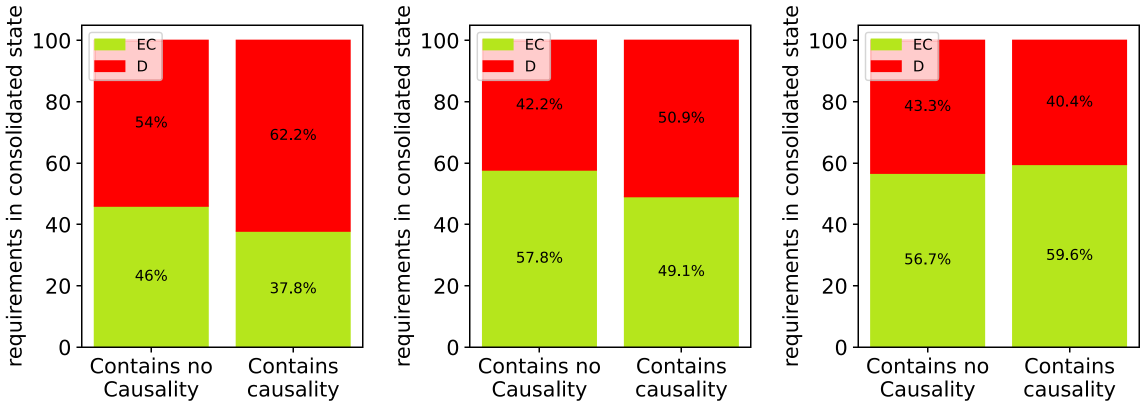
(a) Small requirements **(b)** Medium requirements **(c)** Large requirements

Fig. 8 Distribution of lead-time for binary granularity split by the size of requirements (H_{1_0})

Fig. 9 Distribution of filtered consolidated states (H_{2_0})



(a) Binary granularity G1 **(b)** Batch granularity G2



(a) Small requirements **(b)** Medium requirements **(c)** large requirements

Fig. 10 Distribution of filtered consolidated for binary granularity split by the size of requirements (H_{2_0})

suggests considering causality as an impact factor when estimating the life-cycle of a requirement. A consequent hypothesis of this correlation is that the strict semantic structure of the relation causes an effect on the comprehensibility of a requirement, which makes it easier to translate into downstream artifacts like code or test cases.

Answer to RQ 8: The use of causality does not correlate with the consolidated state of requirements. It is safe to assume that the occurrence of causality does not impact the life-cycle of requirements regarding its consolidated state statistically significant in comparison with other factors.

Answer to RQ 9: The use of causality correlates slightly with smaller volatility of requirements. Comparably to RQ 7, the hypothesis that the semantic structure of the relation causes an effect on the understandability of a requirement, which in turn requires fewer decisions due to being less

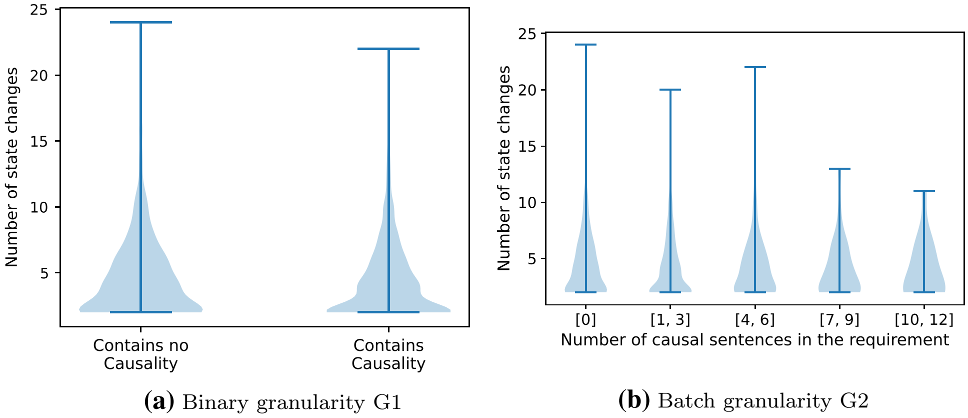


Fig. 11 Distribution of volatility (H_{3_0})

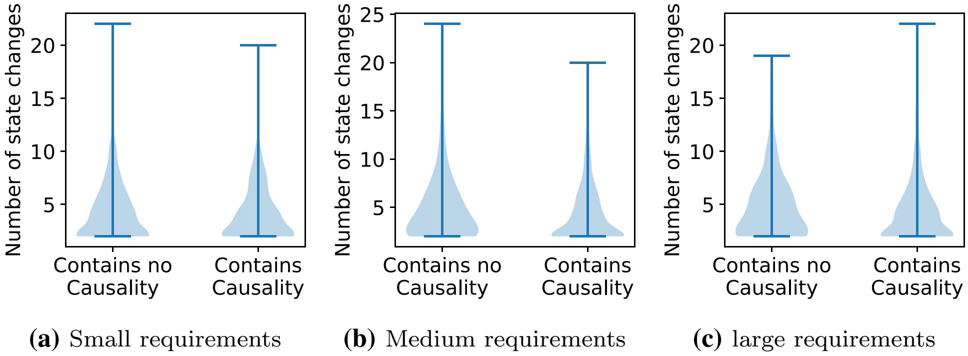


Fig. 12 Distribution of volatility for binary granularity split by the size of requirements (H_{3_0})

ambiguous, can be derived from the correlation analysis. We conclude that the relationship between the use of causality in NL requirements and lead-time as well as the volatility of requirements is worth for further, more thorough investigation: the slight correlation supports the feasibility of considering causality as an aspect of requirements quality.

Applicability of automatic causality detection The initial, exploratory investigation of this phenomenon demonstrates a possible use case of automatic causality detection as part of a quality metric. The small extent of the correlations and their low effect size according to the applied measures emphasize that the occurrence of causality is definitely not the only or most impactful, but certainly a considerable factor for the features of requirements. Considering the detection of causality with the approach presented in this research endeavor as a complement to other requirements quality frameworks such as for example requirements smells [12] might benefit the reliability of these quality metrics by taking positive effects on requirements into account. Future studies need to investigate this claim in further detail.

5.5 Threats to validity

External validity The generalizability of results cannot be claimed based on the exploratory case study on one data set. Further data sets are necessary to be investigated and compared to compensate for context factors like the size and domain of the company, the utilized development process, techniques employed in the requirements engineering phase, as well as applied technologies. However, the analyzed data-set represents five years of product development (between 2010 and 2015) with 41 products and 36 software releases. Therefore, the heterogeneity of authors and editors of requirements is high.

Internal validity To ensure that the correlation between the occurrence of causality and the lead-time of requirements is indeed causal and not confounded, further qualitative analysis beyond the data recorded in the respective data set must be performed. The impact of causality on comprehensibility in contrast to other factors of ambiguity has to be addressed in future studies of qualitative nature. Apart from that, another possible threat to validity is that the analyzed data could contain incorrect information, caused for example by a lack of diligence when providing certain information for a requirement. Since this data set is based on real project data and has been fostered over the course of five years in an industrial setting, the threat is considered low, but still worth mentioning.

6 Related work

6.1 Application of causality detection

As indicated in Sect. 2, many disciplines have already dealt with the notion of causality and explored use cases for its application. One of the earliest applications of causality detection includes the utilization of causality for question answering. Girju et al. [21] propose an approach using lexico-syntactic patterns within one sentence or two adjacent sentences, where the patterns consist of two noun phrases (NP) connected with a causative verb (VP) in the following structure:

$$< NP_1 verb NP_2 > \tag{4}$$

The patterns were built by traversing WordNet concepts for noun phrases that are connected by a *cause-to*-relationship, which is explicitly annotated in the WordNet corpus. Subsequently, from a large NL corpus, all verbs connecting these causally related noun phrase pairs were extracted as causation-verbs. Based on this information and further semantic features from WordNet, the lexico-syntactic patterns detecting causal relations were created. Chang et al. [5] expand on this concept by taking into account conceptual pair probability and cue phrase probability as additional indicators for the classification of a causal relation. The focus on extracting causal relationships to automatically answer why-questions is expanded to the inter-sentential level by Pechsiri et al. [42], who utilize the coexistence of causative and effective verbs as indicators for causal relationships. Other early approaches are rooted in the medical domain, where relationships between symptoms and diseases are commonly expressed in natural language sentences utilizing causality: Khoo et al. [30] extract causal knowledge from a medical database using graphical patterns. The roles and attributes of a causal situation are structured in a three-layer template, which constitutes the framework for manually elicited patterns. More recent approaches like the one proposed by Doan et al. [9] utilize POS tags and dependency parse trees to identify causal relations based on a manually generated set of patterns from a large data set of tweets. In the field of economics, causality detection has been applied to improve the reasoning about market-related relationships. Early approaches include Chan et al. [4] utilizing a hierarchy of manually generated semantic, sentence, and consequence and reason templates. Other approaches like proposed by Inui et al. [28], which also extract causal relations from newspapers, base their causality detection algorithm on the occurrence of cue phrases. The typology defined in the course of this research classifies causal relations with respect to their arguments' *volitionality*, where the volitionality of an event distinguishes an action from a state of affairs. The resulting binary combinations of events of different volitionality constitute the four relationships *cause*, *effect*, *precond*, and *means*. Recent work by Xu et al. [56] acknowledges the lack of focus regarding labeling and extraction methods in the area of causality extraction and contributes by summarizing and evaluating existing causality data sets. Further applications of causality detection include extrapolating causal relations based on semantic relations between nouns [26], effectively increasing the domain of reasoning based on causal relationships.

6.2 Causality in requirements engineering

To the best of our knowledge, we are the first to focus on causality from the perspective of Requirements Engineering. In our previously published papers, we motivated why the RE community should engage with causality [15] and provide empirical evidence for the relevance of causality in requirement documents as well as further insights into its form and complexity [14]. The latter work is extended in the manuscript at hand with an additional, exploratory investigation of the implications of the use of causality in requirements artifacts. Detecting causality in natural language has been investigated by several studies which usually belong to one of two categories according to Asghar et al. [1]: early approaches [30, 55] use handcrafted, lexico-syntactic patterns to identify causal sentences. These approaches are highly dependent on the manually created patterns and show weak performance, inhibiting an effective application in practice as shown in our comparison of algorithms in Sect. 4. Opposed to pattern-matching are feature-based classification methods: recent papers apply neural networks and exploit—similarly to our approach—the Transfer Learning capability of BERT [31]. However, we see a number of problems with these papers regarding the realization of our described RE use cases: First, neither the code nor a demonstration is published, making it difficult to reproduce the results and test the performance on data from the RE domain. Second, they train and evaluate their approaches on strongly unbalanced data sets with causal to non-causal ratios of 1:2 and 1:3, but only report the macro-Recall and macro-Precision values and not the metrics per class. Thus, it is not clear whether the classifier has a bias toward the majority class or not.

7 Conclusions and future work

The behavior of systems is often specified in terms of causal relations in natural language requirements. Efficiently extracting this causal information would allow for effective support of downstream activities that rely on such causal relations, such as the tool-supported derivation of test cases and further activities that need to reason about requirement dependencies [15]. However, contemporary methods still fail to extract causality with reasonable performance [1]. Therefore, we have argued for the need for a novel method for causality extraction and closed this gap with the contributions in this manuscript. We understand causality extraction as a two-step problem: First, we need to detect if requirements have causal properties. Second, we need to comprehend and extract their causal relations. At present, however, we lack knowledge about the form and complexity of causality in requirements, which is needed to develop suitable approaches for these two problems. In this manuscript, we reported on how we addressed this research gap by contributing: (C 1) an exploratory case study with 14,983 sentences from 53 requirements documents originating from 18 different domains. We found that causality is a widely used linguistic pattern to describe system functionalities and that it mainly occurs in explicit, marked form. (C 2) CiRA as an approach for the automatic detection of causality in requirements documents. This constitutes the first step toward causality extraction from NL requirements. We empirically evaluate our approach and achieve a macro-F_1 score of 82 % on real-world data. (C 3) A demonstration of a possible use case of the automatic causality detection approach in a correlation analysis between the occurrence of causality and the life-cycle features of a requirement. (C 4) Finally, by following the open science norms and principles established in the empirical software engineering research community [38], we have further disclosed our entire source code, tool, and annotated data set within the limitations of existing non-disclosure agreements in order to actively support the research community working on same or similar problems and further facilitate independent replications.

Two further research directions are, in our opinion, worth being mentioned here: First, extending the first case study and analyzing the sentences from the requirements documents in a more granular way by categorizing them—e.g., in functional and non-functional requirements—would enrich our current insight into causality in requirements documents in general with further insights into causality in specific requirement categories. This includes investigating the particularities of specific domains, for example to explain the difference in cue phrase precision. Second, tackling the second of the two earlier mentioned sub-problems—the actual extraction of causal relations from causal sentences—will provide the necessary foundation to enable the various use cases. We are currently enhancing our previous approaches [16, 17] with the insights gained from this study and cordially invite the RE community to join the endeavor. Building on the second case study presented in Sect. 5, future studies may continue exploring the relationship between the occurrence of causality and features of requirements. Extending the automatic causality detection approach beyond the current intra-sentential limitations may for example enable to investigate the relationship between requirements' dependencies and features of requirements.

Appendix A Detailed distribution of labels

See Table 11

Table 11 Distribution of labels among all categories for all domains

Domain	Causal		Explicit		Marked		SingleCause		SingleEffect		EventChain		SingleSen		Temporality			Relationship			Sentences
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	before	overlap	during	cause	enable	prevent	
Aerospace	3846	1664	193	1471	296	1368	334	1330	380	1284	1534	130	1550	114	919	610	135	577	268	21	5510
Agriculture	238	57	4	53	1	56	12	45	22	35	55	2	56	1	33	20	4	25	5	1	295
Astronomy	133	106	5	101	27	79	21	85	27	79	94	12	102	4	34	67	5	40	13	3	239
Automotive	119	38	1	37	0	38	15	23	9	29	35	3	38	0	24	13	1	18	2	1	157
Banking	237	139	9	130	19	120	18	121	18	121	127	12	133	6	43	86	10	44	18	0	376
Data Analytics	2028	671	56	615	129	542	105	566	111	560	634	37	629	42	345	266	60	177	62	6	2700
Digital Library	143	55	7	48	2	53	8	47	9	46	54	1	51	4	23	27	5	12	10	0	198
Digital Society	63	44	8	36	14	30	5	39	3	41	41	3	41	3	0	39	5	12	6	0	107
Energy	19	3	0	3	1	2	0	3	0	3	2	1	3	0	1	2	0	2	0	0	22
Health	784	378	26	352	56	322	58	320	82	296	349	29	353	25	167	169	42	105	50	1	1164
Infrastructure	498	278	26	252	38	240	58	220	48	230	262	16	256	22	140	118	20	100	12	4	776
Insurance	11	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	0	0	12
Military	43	1	0	1	0	1	0	1	0	1	1	0	1	0	0	1	0	1	0	0	44
Physics	140	91	5	86	5	86	26	65	21	70	83	8	90	1	34	47	10	37	10	0	232
Regulatory	25	2	0	2	0	2	0	2	0	2	2	0	2	0	1	1	0	1	0	0	27
Smart City	1168	346	48	298	27	319	83	263	79	267	317	29	307	39	131	172	43	107	50	6	1514
Sustainability	848	184	36	148	11	173	36	148	42	142	180	4	161	23	81	84	19	85	23	2	1032
Telecomm	421	157	22	135	23	134	26	131	27	130	154	3	153	4	67	79	11	53	23	3	578
Sum	10764	4215	446	3769	649	3566	805	3410	878	3337	3925	290	3927	288	2044	1801	370	1397	552	48	14983

Acknowledgements We would like to acknowledge that this work was supported, in parts, by the KKS foundation through the S.E.R.T. Research Profile project at Blekinge Institute of Technology. Further, we thank Yannick Debes and Michael Dorner for stimulating discussions and their valuable feedback on earlier versions of this work.

Funding Open access funding provided by Blekinge Institute of Technology.

References

1. Asghar N (2016) Automatic extraction of causal relations from natural language texts: a comprehensive survey. arXiv:1605.07895
2. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Royal Stat Soc Series B (Methodol) 57(1):289–300
3. Blanco E, Castell N, Moldovan D (2008) Causal relation extraction. In: LREC'08
4. Chan K, Lam W (2005) Extracting causation knowledge from natural language texts. Int J Intell Syst 20(3):327–358
5. Chang DS, Choi KS (2004) Causal relation extraction using cue phrase and lexical pair probabilities. In: International Conference on Natural Language Processing, pp. 61–70. Springer
6. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20:37–46
7. Cohen J (1973) Eta-squared and partial eta-squared in fixed factor anova designs. Educ Psychol Measurement 33(1):107–112
8. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
9. Doan S, Yang EW, Tilak SS, Li PW, Zisook DS, Torii M (2019) Extracting health-related causality from twitter messages using natural language processing. BMC Med Inf Decis Mak 19(3):71–77
10. Fares M, Kutuzov A, Oepen S, Velldal E (2017) Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In: NoDaLiDa'17
11. Feinstein AR, Cicchetti DV (1990) High agreement but low kappa: I. the problems of two paradoxes. J Clin Epidemiol 43:543–549
12. Femmer H, Mendez D, Wagner S, Eder S (2017) Rapid quality assurance with requirements smells. J Syst Softw 123:190–213
13. Fischbach J, Frattini J, Mendez D, Unterkalmsteiner M, Femmer H, Vogelsang A (2021) How do practitioners interpret conditionals in requirements? arXiv preprint arXiv:2109.02063
14. Fischbach J, Frattini J, Spaans A, Kummeth M, Vogelsang A, Mendez D, Unterkalmsteiner M (2021) Automatic detection of causality in requirement artifacts: the cira approach. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 19–36. Springer
15. Fischbach J, Hauptmann B, Konwitschny L, Spies D, Vogelsang A (2020) Towards causality extraction from requirements. In: RE'20
16. Fischbach J, Vogelsang A, Spies D, Wehrle A, Junker M, Freudenstein D (2020) Specmate: Automated creation of test cases from acceptance criteria. In: ICST'20
17. Frattini J, Junker M, Unterkalmsteiner M, Mendez D (2020) Automatic extraction of cause-effect-relations from requirements artifacts. In: ASE'20
18. Fricker S, Gorschek T, Byman C, Schmidle A (2010) Handshaking with implementation proposals: negotiating requirements understanding. IEEE Softw 27(2):72–80
19. Friedman JH (2017) The elements of statistical learning: data mining, inference, and prediction. Springer, Berlin
20. Fuchs NE, Schwitter R (1995) Controlled natural language for requirements specifications
21. Girju R (2003) Automatic detection of causal relations for question answering. In: Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12, pp. 76–83. Association for Computational Linguistics
22. Girju R, Moldovan DI, et al (2002) Text mining for causal relations. In: FLAIRS conference, pp. 360–364
23. Gordon M, Harel D (2009) Generating executable scenarios from natural language. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 456–467. Springer
24. Gwet K (2020) AgreeStat Analytics (Cloud-based version (AgreeStat360) was used in Sep'). https://www.agreestat.com/
25. Gwet KL (2014) Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters. Advanced Analytics, LLC
26. Hashimoto C, Torisawa K, Kloetzer J, Oh JH (2015) Generating event causality hypotheses through semantic relations. In: Twenty-Ninth AAAI Conference on Artificial Intelligence
27. Honnibal M, Montani I (2020) spaCy NLP library (We use the newest version of the en_core_web_sm model in Sep'). https://spacy.io/
28. Inui T, Inui K, Matsumoto Y (2005) Acquiring causal knowledge from text using the connective marker tame. ACM Trans Asian Language Inf Process (TALIP) 4(4):435–474
29. James G, Witten D, Hastie T, Tibshirani RE (2013) An introduction to statistical learning. Springer, Berlin
30. Khoo CSG, Chan S, Niu Y (2000) Extracting causal knowledge from a medical database using graphical patterns. In: ACL'00
31. Kyriakakis M, Androutsopoulos I, i Ametllé JG, Saudabayev A (2019) Transfer learning for causal sentence detection. arXiv:1906.07544
32. Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics
33. Latta SC, Howell CA, Dettling MD, Cormier RL (2012) Use of data on avian demographics and site persistence during overwintering to assess quality of restored riparian habitat. Conserv Biol 26(3):482–492
34. Lewis D (2013) Counterfactuals. Wiley, Hoboken
35. Mavin A, Wilkinson P, Harwood A, Novak M (2009) Easy approach to requirements syntax (ears). In: 2009 17th IEEE International Requirements Engineering Conference, pp. 317–322. IEEE
36. McHugh ML (2012) Interrater reliability: the kappa statistic. Biochemia Medica
37. McHugh ML (2013) The chi-square test of independence. Biochemia Medica 23(2):143–149
38. Mendez D, Graziotin D, Wagner S, Seibold H (2020) Open science in software engineering. In: Felderer M, Travassos GH (eds) Contemporary Empirical Methods in Software Engineering. Springer, Berlin

39. Mostafazadeh N, Grealish A, Chambers N, Allen J, Vanderwende L (2016) CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In: EVENTS'16
40. Neves M, Ševa J (2019) An extensive review of tools for manual annotation of documents. Briefings in Bioinformatics
41. Olsson T, Wnuk K, Gorschek T (2019) An empirical study on decision making for quality requirements. J Syst Softw 149:217–233
42. Pechsiri C, Kawtrakul A (2007) Mining causality from texts for question answering system. IEICE TRANS Inf Syst 90(10):1523–1533
43. Pustejovsky J, Stubbs A (2012) Natural Language Annotation for Machine Learning—a Guide to Corpus-Building for Applications. O'Reilly Media Inc, Sebastopol
44. Robson C (2002) Real World Research—A Resource for Social Scientists and Practitioner-Researchers. Wiley-Blackwell, Hoboken
45. Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. Empirical Softw Eng 14:131–134
46. Selway M, Grossmann G, Mayer W, Stumptner M (2015) Formalising natural language specifications using a cognitive linguistic/configuration based approach. Inf Syst 54:191–208
47. Sullivan GM, Feinn R (2012) Using effect size—or why the p value is not enough. J Graduate Med Educ 4(3):279–282
48. Sundararaman D, Subramanian V, Wang G, Si S, Shen D, Wang D, Carin L (2019) Syntax-infused transformer and bert models for machine translation and natural language understanding
49. Viera A, Garrett J (2005) Understanding interobserver agreement: the kappa statistic. Family Med 37:360–363
50. Wagner S, Mendez D, Felderer M, Vetrò A, Kalinowski M, Wieringa R, Pfahl D, Conte T, Christiansson MT, Greer D et al (2019) Status quo in requirements engineering: a theory and a global family of surveys. ACM Trans Softw Eng Methodol (TOSEM) 28(2):1–48
51. Wnuk K, Gorschek T, Callele D, Karlsson EA, Åhlin E, Regnell B (2015) Supporting scope tracking and visualization for very large-scale requirements engineering-utilizing fsc+, decision patterns, and atomic decision visualizations. IEEE Trans Softw Eng 42(1):47–74
52. Wolff P (2007) Representing causation. J Exp Psychol General 136:82
53. Wolff P, Song G (2003) Models of causation and the semantics of causal verbs. Cognitive Psychol 47:276–332
54. Wongpakaran N, Wongpakaran T, Wedding D, Gwet K (2013) A comparison of cohen's kappa and gwet's ac1 when calculating inter-rater reliability coefficients: A study conducted with personality disorder samples. BMC Medical Research Methodology
55. Wu CH, Yu LC, Jang FL (2005) Using semantic dependencies to mine depressive symptoms from consultation records. IEEE Intell Syst 20:50–58
56. Xu J, Zuo W, Liang S, Zuo X (2020) A review of dataset and labeling methods for causality extraction. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1519–1531