The final authenticated version is available online at
https://doi.org/10.1007/978-3-031-34560-9_24

Detecting Deviations Between External and Internal Regulatory Requirements for Improved Process Compliance Assessment

Catherine Sai, Karolin Winter, Elsa Fernanda, and Stefanie Rinderle-Ma[0000−0001−5656−6108]

Technical University of Munich, Germany; TUM School of Computation, Information and Technology
{catherine.sai,karolin.winter,elsa.fernanda,stefanie.rinderle-ma}@tum.de

Abstract. In order to assure process compliance, a wide range of regulatory requirements from various documents must be considered. These external requirements are typically transformed into internal requirements such as policies or handbooks for process compliance in an organization. The transformation is mostly done manually, without the ability of a digitalized quality check. To support users, this work provides a semi-automatic approach based on state-of-the-art NLP algorithms. We first provide a list of Regulatory Compliance Assessment Solution Requirements (RCASR) based on which deviations between external and internal textual requirements can be detected and the root cause of the deviations can be identified. This detailed analysis helps to find mitigation actions in order to improve process compliance. The proposed approach is evaluated based on two Case studies with greatly varying regulatory documents and their realizations by companies. The evaluation demonstrates the feasibility of the approach and provides further insights into the applicability of NLP-based automation techniques in the field of process compliance assurance and management.

Keywords: Regulatory Compliance, Natural Language Processing, Alignment Support, Process Conformance

1 Introduction

Implementing regulatory documents is an expensive and cumbersome task for all companies, with severe consequences if their processes turn out to be noncompliant. *"Since the fall of 2021, Ireland's DPC has slapped Meta with 912 million euros in fines, [...] for alleged violations of Europe's signature data privacy law, known as the General Data Protection Regulation (GDPR)."*[1]. Complex regulations such as the GDPR require a lot of expert knowledge to read, understand and finally implement them which is still mostly done manually [15]. An increasing flood of regulatory documents, makes the system-supported implementation

[1] https://edition.cnn.com/2022/11/28/tech/meta-irish-fine-privacy-law/

of regulatory documents more essential for companies than ever. Existing work in this research area has by now mostly focused on, e.g., deriving formalized constraints (cf., e.g., [4]) or process models from natural language text (cf., e.g., [1, 12, 31]). A recent approach assesses compliance between regulatory documents and process models [29]. However, regulatory documents usually need to be contextualized and adapted to a company's environment. For this purpose realizations of regulatory documents, i.e., internal documents such as handbooks or policies are distributed among employees setting out guidelines that need to be adhered to [15]. The goal is to combine the compliance requirements with company specific requirements. Realizations are often the intermediate step between external regulatory requirements and business processes. By ensuring that the formulation in corporate language, i.e. the realization, correctly reflects the external requirements, one lays the foundation that the business processes (which in turn are often aligned with the realization) are also compliant. Although this plays a crucial role for compliance management, machine learning assisted techniques for a compliance degree assessment between regulatory documents and their realizations are currently missing. Hence, this paper aims at providing an approach for this challenging task. As stated in [29], constraints, i.e., sentences containing signal words like "shall, should, must", offer the right level of abstraction to represent the semantics of regulatory documents. By operating on this constraint level, the approach also becomes independent of the order in which constraints are contained in the document. The challenge is to assess *coverage* and at the same time *deviations* between regulatory and realization constraint sets. For this, at first, *Regulatory Compliance Assessment Solution Requirements (RCASRs)* are elicited based on existing work (cf. Sect. 2). Based on the RCASRs, we present a compliance assessment approach (cf. Sect. 3). The first step is to map constraints from regulatory documents with their presumed counterparts from the realization, resulting in a set of constraint pairs. The constraint pairs are analyzed for different deviations, e.g., responsibility deviations, i.e., whether a task is executed by the correct resource. The results of this deviation analysis are aggregated in order to derive an overall compliance degree between a regulatory document and a realization. The approach is implemented and evaluated based on two real-world case studies with the GDPR and an ISO Norm (cf. Sect. 4). A discussion of limitations is outlined in Sect. 5. Related work is discussed in Sect. 6, a summary and outlook in Sect. 7 conclude the paper.

2 Regulatory Compliance Assessment Solution Requirements

Comparing regulatory documents and their realizations in a meaningful way is challenging w.r.t. granularity and significance of the comparison. Regarding the granularity, analogously to [29], we operate on *constraints* which constitute an adequate level of abstraction to represent semantics of regulatory documents. Regarding the significance, in the following, we elicit 13 *Regulatory Compliance Assessment Solution Requirements (RCASRs)* (cf. Tab. 1) for a (quantifiable) com-

RCASR #: name	**explanation**	**example**
1: regulatory document relevance	companies and organizations have to identify the various regulatory documents they need to comply with	an international company must comply with regulatory documents from different regions (e.g. EU regulations)
2: content relevance	within the identified documents of RCASR1, the conformance relevant parts for a given company need to be identified	EU GDPR, chapter 7 is not relevant for company x
3: constraint coverage	evaluation, if all constraints from regulatory documents are mentioned in a realization	20 relevant constraints from the EU GDPR are not mentioned in the data protection policy of a company
4: severity deviation	evaluation, if a realization is over-compliant, meaning it is stricter about aspects of constraints or includes constraints that are not required by the regulatory document	the data protection policy of a company states to inform a data subject within 24h, while the GDPR only requires to inform within 72h
5: execution style deviation	within the covered constraints from RCASR3, the phrase referring to how something is supposed to be done deviates between the regulatory document and its realization	the regulatory document requires gluing parts together, the realization states to weld the parts
6: negation deviation	evaluation if the constraint aspects (RCASR5, RCASR7 and RCASR8-10) are similar but negated	the regulatory document requires informing the customer via phone call, the realization states not to reach out via phone.
7: responsibility deviation	within the covered constraints from RCASR3, the phrase referring to who is supposed to be doing something deviates between the regulatory document and its realization	the regulatory document specifies that resource A must execute task t but in the realization, resource B is specified
8: data deviation	within the covered constraints from RCASR3, the phrase referring to what is supposed to be done deviates between the regulatory document and its realization	the regulatory document specifies to consider something as private data, the realization considers it as public data
9: time deviation	within the covered constraints from RCASR3, the time something is allowed to take deviates between the regulatory document and its realization	the regulatory document states a task must be finished within one day, the realization states it must be finished within two days
10: task execution order deviation	within the covered constraints from RCASR3, the order in which actions must be taken deviates between the regulatory document and its realization	the regulatory document states the order of events A-B-C, the realization states the order must be B-A-C
11: constraint duplicates	evaluation, if the same constraint appears multiple times in a realization	the constraint "the data subject needs to be informed of a data breach within 72h by the processor" appears 3 times in a realization
12: overall regulatory compliance	weighted aggregation of the coverage and deviation findings to enable a quantifiable degree of compliance between a regulatory document and its realization	80% of the EU GDPR is correctly covered by the data protection policy of an EU company
13: mitigation actions	provision of recommendations for improving regulatory compliance of a realization	change the order of events to A-B-C as stated in constraint x of the regulatory document

Table 1. Regulatory Compliance Assessment Solution Requirements (RCASRs)

parison. The RCASRs break the task of regulatory compliance assessment down into the steps, necessary to assure quality. Recent solutions comprise [10, 20, 21], comparing regulatory documents and realizations on a document or segment level. Especially in light of explainability and mitigation actions, we aim at a much deeper level of comparison between parts of constraints. This way our solution does not only indicate a deviation on, e.g., chapter level but identifies exactly which part of a constraint sentence deviates. The RCASRs are particularly suited for monitoring compliance, i.e., we do not target the initial modeling of a realization from a regulatory document, but the (continuous) compliance assessment of already modeled realizations for, e.g., checking whether a regulatory document and its realization still comply after changes.

The RCASRs are inspired and extended, based on the work from [3,19,29,30]. [29] compare requirements imposed by regulatory documents to business process models. They propose a cost score containing 3 types of violation that can be observed: 1) missing obligatory activity (compare RCASR 3); 2) wrong resource performing the activity (compare RCASR 7); 3) wrong order of activities (compare RCASR 10). [19] describe 10 compliance monitoring functionalities (CMFs) a holistic approach should address. For our application we found CMF 1–3, as well as CMF 9 and 10 to be relevant: CMF 1: Constraints referring to time (compare RCASR 9); CMF 2: Constraints referring to data (compare RCASR 8); CMF 3: Constraints referring to resources (compare RCASR 7); CMF 9: Ability to explain the root cause of a violation (compare RCASR 13); CMF 10: Ability to quantify the degree of compliance (compare RCASR 12). Finally, [3] introduces the phases of a compliance requirements assessment, starting with the discovery of relevant regulatory documents (compare RCASR 1). We further break this step down in not only identifying the relevant documents (RCASR 1) but also the relevant sections within a document (RCASR 2) and the extraction and completeness assessment of the constraints within the Sections (RCASR 3). [30] present an approach to identify redundant (compare RCASR 11), subsumed (RCASR 4) and in general conflicting (compare RCASR 5-6) constraints in text.

We combine these finding to allow a holistic and more detailed understanding of deviations. RCASR 4 and 11 are concerned with deviations that do not necessarily result in a violation, like a realization containing duplicate mentions of a constraint (RCASR 11), which might not be necessary, or a realization being unintentionally more strict about a constraint than the corresponding regulation (RCASR 4). Such an over-compliance can lead to higher costs and should thus be detected. RCASR 5 is concerned with deviations in how something is supposed to be done. The negation deviation RCASR 6 could have been integrated into RCASR 5, 7, 8 - 10 but as it is a challenging task to reliably identify the correct relation of a negation, we decided to leave it as a separate deviation to identify. This paper addresses RCASR2 – RCASR8 by constraint extraction, determination of constraint coverage and decomposition of a constraint into three parts reflecting the responsibility, task and data aspects. RCASR11 – RCASR13 are partly addressed and RCASR9 and RCASR10 require more complex means,

i.e., time aspects need to be determined which is not a trivial task and in order to extract a control-flow from text we cannot consider constraints isolated from each other [31].

3 Compliance Degree Assessment Approach

As depicted in Fig. 1 the approach takes a regulatory document and its realization as input. The final outcome is a detailed assessment of the overall compliance degree based on constraint coverage and constraint deviation findings (post-processing). For those steps marked with grey boxes (*Relevance Identification*, *1st Step Constraint Coverage*) we provide several options for implementing them in order to cope with, e.g., different styles of realizations. Note that a manual refinement of results is constantly possible, leaving room for incorporating users in the overall compliance degree assessment during each step.

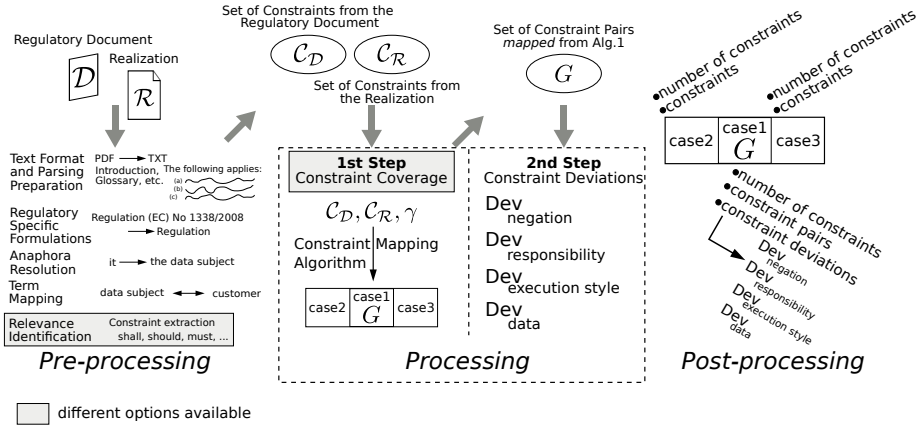

Fig. 1. Overview of Compliance Degree Assessment Approach

3.1 Pre-processing

The pre-processing is performed in a semi-automated manner with multiple cyber-human interaction options. As initial step, we automate the PDF to plain text transformation but strongly encourage a manual review in order to comprehensively tackle the challenge of sentence boundary detection, cf., e.g., [26]. Fully automated components are the anaphora resolution to enhance that sentences are self-contained and the identification of relevant sentences. For the regulatory document relevance is identified by signal words, concerning the realization, the user can choose to do the same or include all sentences in the assessment. The responsibility mapping again allows for human intervention: an automated extraction of entities facilitates a manual mapping inspection for responsibility terms from a realization and a regulatory document.

3.2 Processing

The processing phase consists of 2 steps. In step 1, we map similar constraints from realization and regulatory document by 5 different options. For the 2nd step, a decomposition of constraints into parts containing the responsibility information, execution style, data and negation is carried out (a) and the parts similarity is computed (b).

1st Step – constraint coverage. The pre-processing resulted in the identification of relevant content in form of constraints (**RCASR2**). In the following, let $\mathcal{C}_\mathcal{D}$ be the set of constraints from regulatory document $\mathcal{D}$ and $\mathcal{C}_\mathcal{R}$ the set of constraints from the realization $\mathcal{R}$ respectively. In order to assess the compliance degree, associated constraints must be identified first which is a challenging task since regulatory documents and their realizations can be complex. Yet, only if constraints are correctly mapped, we can draw conclusions about constraint deviations in the latter. Algorithm 1, describes the constraint mapping. It takes as input $\mathcal{C}_\mathcal{D}, \mathcal{C}_\mathcal{R}$ and a threshold γ and first of all creates all possible pairs $(c_d, c_r) \in \mathcal{C}_\mathcal{D} \times \mathcal{C}_\mathcal{R}$ ($\mapsto$ line 1). Afterwards, in lines 4–15 those pairs are checked for whether their similarity $sim(c_d, c_r)$ is above threshold γ and if so, the pair is added to the list of mapped pairs.

Algorithm 1 Constraint Mapping Algorithm

Input: $\mathcal{C}_\mathcal{D}, \mathcal{C}_\mathcal{R}, \gamma$
Output: not_mapped, mapped

```
1: pairs = create_pairs(C_D, C_R)
2: mapped = {}
3: not_mapped = {}
4: for pair in pairs do
5:     if sim(pair) < γ then
6:         continue
7:     end if
8:     if mapped[pair[0]] == None then
9:         mapped[pair[0]] = {"realization": pair[1], "sim": sim(pair)}
10:    else
11:        if sim(pair) > mapped[pair[0]].sim then
12:            mapped[pair[0]] = {"realization": pair[1], "sim": sim(pair)}
13:        end if
14:    end if
15: end for
16: for pair in pairs do
17:    if pair[0] not in mapped.keys then
18:        not_mapped["regdoc"] << pair[0]
19:    end if
20:    if pair[1] not in mapped.values.map(val → val["realization"]) then
21:        not_mapped["realization"] << pair[1]
22:    end if
23: end for
```

Next, new pairs are filtered such that one constraint from the realization can have multiple counterparts in a regulatory document, but one constraint from a regulatory document can only have one counterpart within the realization by only retaining pairs with maximal similarity. The last part of the algorithm, lines 16–23, determines all components of pairs that were not mapped at all whereas we distinguish between not mapped constraints from a regulatory document and the ones not mapped from a realization. For the mapping, i.e., calculating the similarity of constraint pairs (*sim(pairs)*), the natural language text is first transformed in a mathematical representation, called vector or embedding. The similarity between these embeddings can be calculated by various approaches, the cosine similarity being most commonly used. The cosine similarity is based on the cosine of the angle between two vectors [14]. We provide the following five options of embedding methods for the implementation.

S-BERT sentence transformer embeddings and cosine similarity [2]: chosen as transformers are state-of-the art in NLP since their introduction in 2017 [13]. S-BERT is a transformer model specifically trained on estimating the similarity of sentences and comes with a significant performance increase compared to using BERT for the same task [22].

Legal-S-BERT legal sentence transformer embeddings and cosine similarity [8]: implemented to evaluate how a transformer model trained on the same domain as the application will affect the results. Legal-BERT is a transformer model trained on a wide span of legal texts [8]. In order to apply Legal-BERT as Legal-S-Bert, we performed a domain adaption [23], training a new model with Legal-BERT as pre-training on the target domain and the Natural Language Inference (AllNLI) dataset [5, 28] for fine-tuning on labeled data.

TM topic modeling, word2vec and cosine similarity [32]: with this approach we test the hypothesis, that first clustering the constraints will improve the similarity identification. Within clusters including both regulatory and realization constraints, we calculate the similarity of the pairs as described in Alg. 1. For Topic Modeling we use Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) which is a topic modeling approach with the underlying assumption of one topic per document [32]. This makes it especially useful for short documents or single sentences as we consider in this case.

k-means k-means, word2vec and cosine similarity: this approach is similar to **TM** as it groups the constraints before calculating similarity between the constraints in one group. Clustering is performed by first embedding the constraints as vectors by means of the sentence transformer model and then applying k-means to assign each constraint to a cluster.

key phrases key phrase extraction, S-BERT embedding and cosine similarity [24]: in contrast to the other approaches, here we do not compare the whole constraint text but extract key phrases from the text and section title to calculate the similarity between these key terms of the regulatory and re-

alization constraints. The key phrase extraction is implemented with RAKE (Rapid Automatic Keyword Extraction) algorithm [24].

Based on the result of Alg. 1 we can distinguish three cases. Case 1 is a set of constraint pairs which were mapped (**RCASR3**). Case 2 is a set of constraints from the regulatory document having no counterpart in the realization and case 3 the set of constraints within the realization having no counterpart in the regulatory document. Case 2 is particularly interesting because it could be an evidence that the realization is missing out some regulatory document parts. However, only a manual inspection can reveal whether these parts should have been included. Therefore, they will be extracted for a stakeholder review and we will not consider these constraints when determining constraint deviations. Case 3 indicates that the realization contains company specific parts that are not relevant in the regulatory context. We do not consider those constraints further, but they do give an indication as to topics the company is executing more severely than they are required to (**RCASR4**). Based on these observations, a notion for the **constraint coverage degree** is given as follows:

- Full constraint coverage of a regulatory document and a realization holds if all constraints were mapped by Alg. 1.
- Partial constraint coverage of a regulatory document and a realization holds if *not_mapped* is not empty after applying Alg. 1
- No constraint coverage of a regulatory document and a realization holds if *mapped* is empty after Alg. 1

2nd Step – constraint deviations. In the second step of the processing phase, we address determining constraint deviations. As already stated, for the compliance degree assessment only constraint pairs from case 1 can be considered for a deviation analysis. Let in the following G denote the set of all constraint pairs in *mapped* after applying Alg. 1. In order to derive deviations, we decompose each constraint by utilization of custom built functions in combination with the "Part of Speech" (POS) information, specific occurrence matching, as well as position (e.g. subtree) and dependencies of a word in its constraint context. The outcome are phrases reflecting the *responsibility*, *task* and *data* parts, e.g., $c =$"*A data officer must glue all product parts.*" is decomposed into $c|_{res} =$"*data officer*", $c|_{task} =$"*glue*", $c|_{data} =$"*all product parts*". Let $c|p$ denote the decomposition of $c \in \mathcal{C}_\mathcal{D} \cup \mathcal{C}_\mathcal{R}$ onto its part $p \in \{\text{res, task, data}\}$. Based on this decomposition the deviations for RCASR 5–8 can be defined using again the cosine similarity for comparing the parts as follows.

Definition 1 (execution style deviation – RCASR5). *An execution style deviation for a constraint pair* $(c_d, c_r) \in G$ *occurs if for thresholds* $\gamma_i \in [-1, 1]$, $i = 1, 2, 3$*:*

$$sim(c_d|_{res}, c_r|_{res}) \geq \gamma_1 \wedge sim(c_d|_{task}, c_r|_{task}) < \gamma_2 \wedge sim(c_d|_{data}, c_r|_{data}) \geq \gamma_3$$

A deviation in execution style occurs whenever the similarity between two tasks is below a threshold but the responsibility and data parts are above a

threshold. Consider, e.g., for $\gamma_1 = \gamma_2 = 1$ and $\gamma_2 =$ the constraints $c_d =$ *"A worker must glue all product parts."* and $c_r =$ *"A worker must weld all product parts"*. Then $sim($*"worker"*, *"worker"*$) = 1 = sim($*"all product parts"*, *"all product parts"*$)$ but $sim($*"glue"*, *"weld"*$) < \gamma_2 = 0.17$.

Definition 2 (negation deviation – RCASR6). *A negation deviation for a constraint pair $(c_d, c_r) \in G$ occurs if for thresholds $\gamma_i \in [-1, 1], i = 1, 2, 3$ holds $sim(c_d|_{res}, c_r|_{res}) \geq \gamma_1 \wedge sim(c_d|_{task}, c_r|_{task}) \geq \gamma_2 \wedge sim(c_d|_{data}, c_r|_{data}) \geq \gamma_3$ and there exists a negation for either $c_d|_{task}$ or $c_r|_{task}$*

A negation deviation occurs whenever a task within a constraint is negated, but all other parts have similarities above thresholds $\gamma_i, i = 1, 2, 3$, e.g., $c_d =$ *"A data subject shall be informed about a data breach."* and $c_r =$ *"A data subject shall not be informed about a data breach."* Then responsibility, task and data similarities are both equal to 1 but the task is negated.

Definition 3 (responsibility deviation – RCASR7). *A responsibility deviation for a constraint pair $(c_d, c_r) \in G$ occurs if for thresholds $\gamma_i \in [-1, 1]$, $i = 1, 2, 3$ holds:*
$sim(c_d|_{res}, c_r|_{res}) < \gamma_1 \wedge sim(c_d|_{task}, c_r|_{task}) \geq \gamma_2 \wedge sim(c_d|_{data}, c_r|_{data}) \geq \gamma_3$

To illustrate the responsibility deviation consider $c_d =$ *"The chief data officer must take care of a data breach."* and $c_r =$*"The data officer must take care of a data breach."*. As $sim($"chief data officer", "data officer"$) = 0.9$ is very high, this deviation will only be detected with a $\gamma_1 > 0.9$.

Definition 4 (data deviation – RCASR8). *A data deviation for a constraint pair $(c_d, c_r) \in G$ occurs if for thresholds $\gamma_i \in [-1, 1], i = 1, 2, 3$ holds:*
$sim(c_d|_{res}, c_r|_{res}) \geq \gamma_1 \wedge sim(c_d|_{task}, c_r|_{task}) \geq \gamma_2 \wedge sim(c_d|_{data}, c_r|_{data}) < \gamma_3$

A data deviation can occur if the responsibility and task similarity are above thresholds γ_1 and γ_2, but the data similarity is below γ_3. An example for that case is $c_d =$*"A data officer should process public data."* and $c_r =$*"A data officer should process private data."*. Again, task as well as responsibility similarities are equal to 1 but $sim($"public data", "private data"$) < \gamma_3 = 0.81$.

Note that, just because none of the deviation definitions holds for a constraint pair, there can still be deviations. Consider, e.g., a constraint pair (c_d, c_r) with $sim(c_d|_{res}, c_r|_{res}) < \gamma_1 \wedge sim(c_d|_{task}, c_r|_{task}) \geq \gamma_2 \wedge sim(c_d|_{data}, c_r|_{data}) < \gamma_3$. In this case neither Def. 3 nor Def. 4 holds but still, the responsibility as well as data aspects do not seem to be correct. Therefore, only a manual inspection of constraint pairs having no deviations can reveal the final overall deviation result. We can only provide deviation statements for unambiguous cases.

Analogously to constraint coverage, notions for the **degree of constraint deviations** can be defined based on the definitions for the deviation types as follows:

1) Full constraint deviation of a regulatory document and a realization holds if for all $(c_d, c_r) \in G$ at most one deviation definition holds. 2) Partial constraint

deviation of a regulatory document and a realization holds if for at least one $(c_d, c_r) \in G$ at least one deviation definition holds. 3) No constraint deviation of a regulatory document and a realization holds if for all $(c_d, c_r) \in G$ no constraint deviation definition holds.

3.3 Post-processing

Based on the findings in Sect. 3.2 we can now assess the overall compliance degree of a regulatory document and a realization. In particular, as depicted in Fig. 1 a detailed analysis on how many and which constraints from a realization, resp. regulatory document were (not) mapped can be provided. Not mapped constraints are considered as not covered by a realization (RCASR3) or not required by a regulatory document (RCASR4). For the group G of mapped constraints, we can provide insights on the number and concrete constraints for which one of the defined constraint deviations (RCASR5–8) occurred. This provides users with appropriate means to identify potential compliance violations or over-compliance. Within the extracted cases 1–3, constraint duplicates can be easily identified (RCASR11). Through the ratio of constraints in case 1 compared to case 2&3 and the count of deviations found, the overall regulatory compliance can be assessed (RCASR12) and the detailed information about the deviation cause aids the mitigation actions (RCASR13). This detailed analysis of a constraint's components not only recognizes a potential constraint deviation, but also indicates the cause of the deviation, which leads to better stakeholder acceptance of the results and aids the improvement of the deviating constraints.

4 Evaluation

The approach is prototypically implemented[2] in Python 3 using (i.a.) the NLP framework spaCy [16]. The approach was implemented for two case studies. Case study 1 was evaluated in a qualitative and quantitative manner and will be discussed in detail in Sect. 4.1 – 4.3. For Case study 2 we did not create a gold standard due to missing expertise in the field of this regulatory document. Thus, only a qualitative evaluation with regards to cross-regulation application was performed and will be introduced in Sect. 4.4.

4.1 Pre-processing

In Case study 1, the compliance degree of the General Data Protection Regulation (GDPR)[3] and a company's "Data Protection Policy EU" is assessed. For the GDPR articles 1 to 4 and articles 92 to 99 are excluded since they only contain information on when to apply the regulation and a glossary of terms. For the company's policy all text passages are taken into account. Overall, the automatic approach retrieved 423 regulatory constraints, 120 realization constraints with signal words and 264 when taking all sentences from the realization.

[2] `https://github.com/CatherineSai/compliance_textual_constraints`
[3] `https://gdpr.eu/tag/gdpr/`

4.2 Processing

To evaluate Step 1, we manually derived gold standards for case 1 with both options of *Relevance Identification*, i.e., i) using signal words within the realization and ii) using no signal words respectively. Team members independently created a mapping of regulatory-realization-constraint-pairs which were discussed resulting in two gold standards. The gold standard for i) consists of 56 matches while ii) consists of 88 matches. This indicates that selecting ii) is the preferable option as it delivers more constraint pairs than option i). This initial gold standard was later enhanced as further reviews showed, that often multiple realization constraint can individually be a sufficient match for a given regulatory constraint. The enhanced gold standard for step one therefore includes all sufficient regulatory-realization-constraint-pairs. The gold standard for Step 2 was created in the same manner based on the gold standard of Step 1.

Step 1. Keeping in mind NLP challenges like ambiguity, context dependent meaning or implicit (human common) knowledge, it is challenging to implement an automated approach that reaches the exact same understanding of complex regulatory texts as a human. For the intended application of our approach, we thus focused on the recall, i.e., how many of the in the gold standard defined regulatory-realization-constraint-pairs were identified (cf. Tab. 2). The relatively low precision in Tab. 2 means that, e.g., for the enhanced Legal-S-BERT the users still have to review 285 proposed pairs for case 1 from the model. Additionally, 175 potentially missing regulatory constraints were identified as case 2 and 186 as case 3. However, 423 regulatory constraints and 264 realization constraints lead to over 100000 possible combinations, so our approach is a major improvement.

	key phrases (initial)	S-BERT (initial)	L-S-BERT (enhanced)
γ	0.74	0.7	0.72
% precision	12.5	16.9	20.1
% recall	12.5	30.7	70.6

Table 2. Step 1 Results best three methods GDPR Case Study

The initial $\gamma-$setting for Alg. 1, was selected after reviewing all highest similarity scores of constraint pairs. Choosing similarity of $\gamma = 0.7$ results in sufficient content-related pairs. Below $\gamma = 0.7$ there are too few content-related constraint pairs. The γ differ across methods because the embeddings and thus the similarity score between the models differ. The selected γ balances the amount of false positive and false negative matches.

Step 2. For the second step of the deviation assessment, we analyzed the 88 identified gold standard constraint matches in further detail. This way Step 2 can be evaluated without influence of the matching performance in Step 1. Initially, for Step 2.a), the constraints are decomposed into the four parts defined in Sect. 3.1. The results can be seen in Tab. 3. This is followed by Step 2.b), the similarity comparison between the identified constraint parts.

	Responsibility	Execution Style	Data	Negation
reg.	95.5	89.7	80.5	83.3
rea.	94.7	87.7	75.5	100

Table 3. Step 2.a) Results GDPR Case Study, % accuracy

	Resp.	Exec. Style	Data
γ	$\gamma_1 = 0.39$	$\gamma_2 = 0.34$	$\gamma_3 = 0.3$
% acc.	92	87.5	97

Table 4. Step 2.b) Results GDPR Case Study

For the selection of γ_1–γ_3, the variability of the phrases needs to be considered (e.g. responsibility is usually concise with 1-2 words that occur often like "controller", the data phrase on the other hand is likely to cover a longer span of words with higher variance). This also leads to the high accuracy of the similarity computation for the data component: as the text spans are much longer, the similarity computation is less dependent on each word. With the setting in Tab. 4, we retrieved 11 execution style, 0 negation, 17 responsibility and 1 data deviation.

4.3 Post-processing and Findings

By reviewing the results of Steps 1 and 2, we observed the following: i) the constraint is truly missing in the realization and thus a violation ($\mapsto$ Case 2), ii) multiple deviation definitions hold leading to ambiguous cases that need manual resolution, iii) the constraints are very dissimilar which cause an erroneous automatic analysis. The 17 identified responsibility deviation are caused by three reasons: false similarity calculation and thus a false positive deviation, an unspecific subject in the realization (e.g., "the responsible department" or "management", which could be a controller, processor, or third party vs. "the controller"), or due to the acting role being missing in the constraint as it is formulated in passive style. The data deviation is due to wrong object phrase parsing and the execution style deviations are caused by false similarity calculation and different writing style (e.g. from "obtain [...] the erasure", only "obtain" is the execution style and thus causes a deviation if compared to "be deleted"), these are considered false positives and thus not true deviations. As the unspecific and missing responsibilities are true deviations, we have an overall precision for the deviation detection of 55%. The results of Tab. 2-4 show, that the approach and technologies used work. However, some of the constraints and their parts being compared are still different (which is especially visible in the true positive execution style deviations) and thus need to be made more comparable (see Sect. 5 Comparability) in the pre-processing to improve the deviation assessment. Thus, this work is just a first step, which does not assess the compliance of all constraints fully automated, but aids a user in directly recognizing the most similar constraint pairs and the deviations within these.

4.4 Cross-regulation Application Through Case Study 2

As second Case study, the regulatory document ISO27001 on information security management systems is compared to "Implementation Guideline ISO/IEC 27001:2013" realization from a company. In the ISO norm, Sections 0 – 3, Annex and Bibliography are excluded as they contain no constraints. Analogously, for the companies guideline, the preamble, Glossary, References, Index and Appendix were not considered. The ISO Case study differs in many aspects from

the GDPR Case study and was analyzed to evaluate the generalizability of the approach. Based on the pre-processing, 139 regulatory constraints and 278 constraints from the realization were extracted. Running the same configuration as in Case study 1, we achieved similar deviation occurrences, leading to the assumption that the choice of thresholds (γ, γ_1–γ_3) can be transferred from one Case study to another without individual selection for each application. This suggests that if our approach was to be applied to a different use case in practise, there would be no need for the creation of a gold standard. The identified thresholds (γ, γ_1–γ_3) can be used as a basis and fine tuned to the specific use case via manual inspection of the results, i.e. the similarity of deviation aspects. Generalizability limitations for Case study 2 are further detailed in Sect. 5.

5 Discussion

This section discusses the generalizability to other regulatory documents and their realizations and the comparability enhancement in order to improve the precision in Step 1 and constraint part deviation assessment in Step 2.

Generalizability. The applicability to other regulatory documents may require an enhancement of the *constraint signal words*. Thus, for Case Study 2 the predefined set of signal words was enhanced with "duties", "requirements", "require" and its conjugations. Additionally, the *choice of similarity thresholds* needs to be reviewed for other use cases as they strongly influence the results.

Adjustments to Alg. 1 become necessary if the realization of the use case is formulated in a way, that multiple realization sentences combined are fulfilling one regulatory constraint. In Case study 1, the regulatory document was more extended, resulting in almost 3 times as many constraints from the regulatory document compared to the realization. Therefore, Alg. 1 is designed to map only one realization sentence to each regulatory constraint. However, in Case study 2, the realization is the more comprehensive document, suggesting that in this Case study a regulatory constraint is represented as multiple realization sentences. Thus Alg. 1 has to be adjusted to allow multiple realization constraints to be mapped to one regulatory constraint.

Comparability. To improve comparability between the two documents, different *levels of abstraction and length* of texts need to be better incorporated in the similarity computations. Transforming *passive formulated sentences to active* writing also poses a major challenge as current solutions do not deliver satisfactory results for long and complex regulatory sentences and can not handle implicit subjects in passive sentences. This is especially important for the deviation assessment in Step 2. Moreover, there is a necessity to *include meta information* such as document structure and referenced content. Regulatory documents and their realizations contain various references to other parts of the same document or other documents, such as *"where the processing is based on point (f) of Article 6(1)"*. Without the information what is contained in *"point (f) of Article 6(1)"*, the constraint is not self-contained and can only be assessed insufficiently.

Additionally, the *inclusion of the documents structure information*, e.g. the section title can improve the meaningfulness of a constraint. The presented approach includes section titles for the key phrase method in step 1 but further investigations how to integrate them in the entire approach are necessary.

6 Related Work

[25] address the compliance between policies and actual scenarios by utilizing a question answering method. [18] train BERT variations with annotated bill pairs to calculate the semantic similarity between bills and the approach compares whole subsections of text, rather than in-depth constraint deviations. Additionally, bills greatly vary from regulatory documents and to the best of our knowledge there is no annotated set available we could use to train such a model for our application area. [7] classify EU laws into topics but perform no comparison calculations, neither between the law nor to their realizations. Within our application use case, [10, 17] use BERT and other NLP methods to measure the compliance of policies with the GDPR. However, these approaches calculate the similarity on segment or document level and state "it would be hard to perform rule-by-rule analysis" [10]. Our approach allows for this in-depth deviation analysis. Another line of research focuses on deriving formalized constraints, e.g., [4] or requirements extraction from natural language text operating in a manual, e.g., [6], or (semi-) automatic way [9, 27, 31].

From a technological point of view compared to string matching approaches like [11], our approach uses more sophisticated text matching and similarity means such as BERT and S-BERT [22] which constitute the current state-of-the art for semantic text similarity [13].

7 Conclusion and Outlook

This paper provides an approach for assessing the compliance degree between a regulatory document and its realization, e.g., a policy. Newly proposed Regulatory Compliance Assessment Solution Requirements (RCASR) build the foundation for a fine-granular assessment of constraint coverage and deviations. The compliance degree assessment approach includes pre- and post- processing steps as well as a processing part where coverage and deviations are determined in an automatic way, still leaving room for users to adapt and control the system. The evaluation demonstrates the importance of finding corresponding constraints in the compared documents (step 1), most accurately extracting their phrase components (step 2) and selecting of the corresponding thresholds for both steps. As future work, we aim to address the RCASRs referring to time and task execution order deviation and plan to provide recommendations on the choice of parameters and support for changes of either regulatory documents or realizations.

Acknowledgement This work has been partly funded by SAP SE in the context of the research project "Building Semantic Models for the Process Mining Pipeline".

References

1. van der Aa, H., Ciccio, C.D., Leopold, H., Reijers, H.A.: Extracting declarative process models from natural language. In: Advanced Information Systems Eng. pp. 365–382 (2019). https://doi.org/10.1007/978-3-030-21290-2_23
2. Antic, Z.: Python Natural Language Processing Cookbook. Packt (2021)
3. Awad, A.M.H.A.: A compliance management framework for business process models. Ph.D. thesis, University of Potsdam (2010)
4. Bajwa, I.S., Lee, M.G., Bordbar, B.: SBVR business rules generation from natural language specification. In: AI for Business Agility (2011)
5. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Empirical Methods in Natural Lang. Proc. pp. 632–642 (2015). https://doi.org/10.18653/v1/d15-1075
6. Breaux, T.D., Antón, A.I.: Analyzing regulatory rules for privacy and security requirements. IEEE Trans. Software Eng. **34**(1), 5–20 (2008). https://doi.org/10.1109/TSE.2007.70746
7. Chalkidis, I., Fergadiotis, M., Androutsopoulos, I.: MultiEURLEX – A multilingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In: Empirical Methods in Natural Lang. Proc. pp. 6974–6996 (2021). https://doi.org/10.18653/v1/2021.emnlp-main.559
8. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: the muppets straight out of law school. CoRR **abs/2010.02559** (2020)
9. Dragoni, M., Villata, S., Rizzi, W., Governatori, G.: Combining natural language processing approaches for rule extraction from legal documents. In: AI Approaches to the Complexity of Legal Systems. pp. 287–300 (2017). https://doi.org/10.1007/978-3-030-00178-0_19
10. Elluri, L., Chukkapalli, S.S.L., Joshi, K.P., Finin, T., Joshi, A.: A BERT based approach to measure web services policies compliance with GDPR. IEEE Access **9**, 148004–148016 (2021). https://doi.org/10.1109/ACCESS.2021.3123950
11. Faro, S., Lecroq, T.: The exact online string matching problem: A review of the most recent results. ACM Comput. Surv. **45**(2), 13:1–13:42 (2013). https://doi.org/10.1145/2431211.2431212
12. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Advanced Information Systems Engineering. pp. 482–496 (2011). https://doi.org/10.1007/978-3-642-21640-4_36
13. Gillioz, A., Casas, J., Mugellini, E., Khaled, O.A.: Overview of the transformer-based models for NLP tasks. In: Comp. Science and Inf. Syst. pp. 179–183 (2020). https://doi.org/10.15439/2020F20
14. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edition. Morgan Kaufmann (2011)
15. Hashmi, M., Governatori, G., Lam, H., Wynn, M.T.: Are we done with business process compliance: state of the art and challenges ahead. Knowl. Inf. Syst. **57**(1), 79–133 (2018). https://doi.org/10.1007/s10115-017-1142-1

16. Honnibal, M., Montani, I.: spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. Unpublished software application. https://spacy. io (2017)
17. Kawintiranon, K., Liu, Y.: Towards automatic comparison of data privacy documents: A preliminary experiment on gdpr-like laws. CoRR **abs/2105.10117** (2021)
18. Kim, J., Griggs, E., Kim, I.S., Oh, A.: Learning bill similarity with annotated and augmented corpora of bills. In: Empirical Methods in Natural Lang. Proc. pp. 10048–10064 (2021). https://doi.org/10.18653/v1/2021.emnlp-main.787
19. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: Functionalities, application, and tool-support. Inf. Syst. **54**, 209–234 (2015). https://doi.org/10.1016/j.is.2015.02.007
20. Müller, N.M., Kowatsch, D., Debus, P., Mirdita, D., Böttinger, K.: On GDPR compliance of companies' privacy policies. In: Text, Speech, and Dialogue. vol. 11697, pp. 151–159 (2019). https://doi.org/10.1007/978-3-030-27947-9_13
21. Qamar, A., Javed, T., Beg, M.O.: Detecting compliance of privacy policies with data protection laws. CoRR **abs/2102.12362** (2021)
22. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Empirical Methods in Natural Lang. Proc. pp. 3980–3990 (2019). https://doi.org/10.18653/v1/D19-1410
23. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Empirical Methods in Natural Lang. Proc. pp. 4512–4525 (2020). https://doi.org/10.18653/v1/2020.emnlp-main.365
24. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. Text Mining: Applications and Theory pp. 1 – 20 (2010)
25. Saeidi, M., Yazdani, M., Vlachos, A.: Cross-policy compliance detection via question answering. In: Empirical Methods in Natural Lang. Proc. pp. 8622–8632 (2021). https://doi.org/10.18653/v1/2021.emnlp-main.678
26. Sanchez, G.: Sentence boundary detection in legal text. In: Natural Legal Lang. Proc. Workshop. pp. 31–38 (2019)
27. Sapkota, K., Aldea, A., Younas, M., Duce, D.A., Bañares-Alcántara, R.: Extracting meaningful entities from regulatory text: Towards automating regulatory compliance. In: Workshop on Requirements Engineering and Law. pp. 29–32 (2012). https://doi.org/10.1109/RELAW.2012.6347798
28. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. In: Human Language Technologies. pp. 1112–1122 (2018). https://doi.org/10.18653/v1/n18-1101
29. Winter, K., van der Aa, H., Rinderle-Ma, S., Weidlich, M.: Assessing the compliance of business process models with regulatory documents. In: Conceptual Modeling. pp. 189–203 (2020). https://doi.org/10.1007/978-3-030-62522-1_14
30. Winter, K., Rinderle-Ma, S.: Detecting constraints and their relations from regulatory documents using NLP techniques. In: OTM Conferences. pp. 261–278 (2018). https://doi.org/10.1007/978-3-030-02610-3_15
31. Winter, K., Rinderle-Ma, S.: Deriving and combining mixed graphs from regulatory documents based on constraint relations. In: Advanced Information Systems Engineering. pp. 430–445 (2019). https://doi.org/10.1007/978-3-030-21290-2_27
32. Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short text clustering. In: Knowledge Discovery and Data Mining. pp. 233–242 (2014). https://doi.org/10.1145/2623330.2623715