AI-enabled Regulatory Change Analysis of Legal Requirements

Sallam Abualhaija[§], Marcello Ceci[§],
Nicolas Sannier, Domenico Bianculli
SnT - University of Luxembourg
Luxembourg
Email: firstname.lastname@uni.lu

Lionel C. Briand[†]
Lero SFI centre for Software Research
and University of Limerick
Ireland
School of EECS, University of Ottawa
Canada
Email: lionel.briand@lero.ie

Dirk Zetzsche, Marco Bodellini
FDEF - University of Luxembourg
Luxembourg
Email: firstname.lastname@uni.lu

Abstract—**Statutory law is subject to change as legislation develops over time – new regulation can be introduced, while existing regulation can be amended, or repealed. From a requirements engineering (RE) perspective, such change must be dealt with to ensure the compliance of software systems at all times. Understanding the implications of regulatory change on compliance of software requirements requires navigating hundreds of legal provisions. Analyzing instances of regulatory change entirely manually is not only time-consuming, but also risky, since missing a change may result in non-compliant software which can in turn lead to hefty fines. In this paper, we propose `MURCIA`, an automated approach that leverages recent language models to assist human analysts in analyzing regulatory changes. To build `MURCIA`, we define a taxonomy that characterizes the regulatory changes at the textual level as well as the changes in the text's meaning and legal interpretation. We evaluate `MURCIA` on four regulations from the financial domain. Over our evaluation set, `MURCIA` can identify textual changes with F$_1$ score of 90.5%, and it can provide, according to our taxonomy, the text meaning and legal interpretation with an F$_1$ score of 90.8% and 83.7%, respectively.**

Index Terms—**Regulatory Change, Prompt Engineering, Natural Language Processing (NLP), Large Language Models (LLMs), ChatGPT, Regulatory Compliance.**

I. Introduction

Requirements engineering (RE) significantly contributes to developing legally-compliant software systems through the elicitation, verification, and maintenance of compliance requirements [1], [2]. Such requirements are based on the interpretation of the statutory law, which is the law that exists through legislation and should be distinguished from common law, which is derived from case decisions [3]. However, regulations are often subject to changes, due to the addition, modification or repeal of acts or provisions [4]. Ensuring the compliance of a system at time t_α does not necessarily ensure the compliance of the system at time t_β. Requirements engineers must thus deal with identifying changes in legal requirements and analyzing their implications on existing compliance requirements to avoid the serious consequences of a system becoming non-compliant at any point in time, which may result in many operations, performed by the system, breaching the law and leading, in turn, to hefty fines. We differentiate throughout the paper between *legal requirements* found in a legislative document [5] versus *compliance requirements* which are the corresponding software requirements documented and implemented in a software system.

The provision of financial services, particularly managing and marketing investment funds, are examples of regulated activities that, in the aftermath of the global financial crisis of 2007-2008 [6], have seen a substantial emphasis on regulatory enforcement aiming at better protecting investors and improving the stability of financial markets. From an RE standpoint, changes in financial regulations can affect the operations of a financial IT system and therefore its compliance.

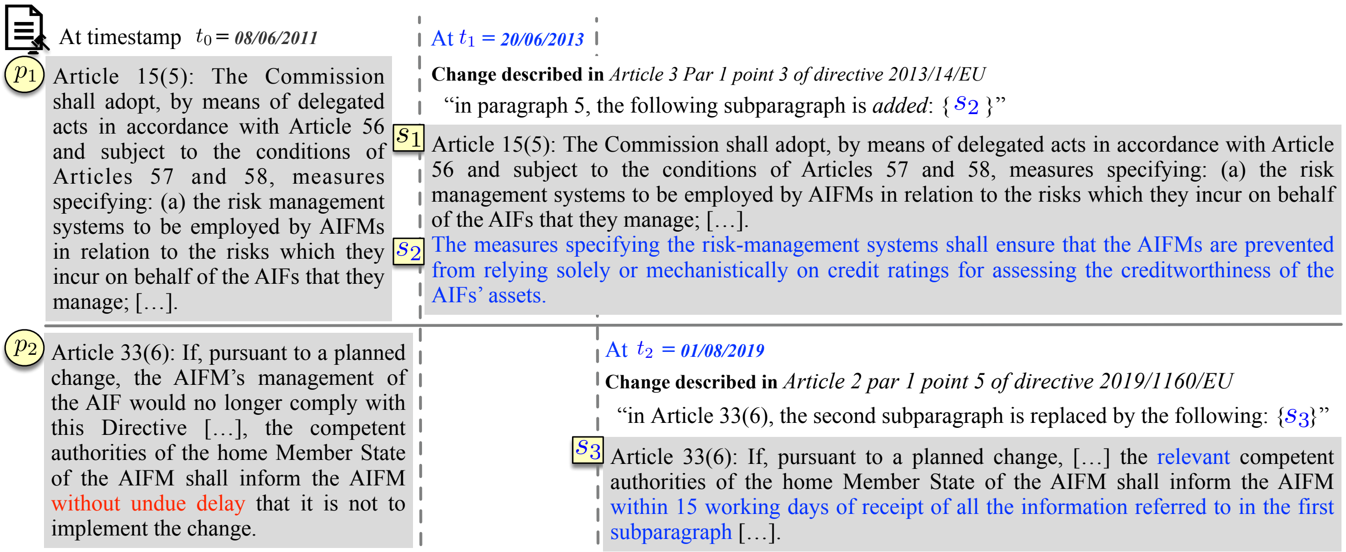
Examples of regulatory changes in the financial domain include the introduction of new legal provisions that modify the criteria for risk calculation [7], prohibit or restrict certain products or services [8], or limit transactions to certain types of assets in specific world regions [9]. Such changes not only entail that the IT system should adjust its operations, e.g., regarding the risk criteria or the investment strategy in certain world regions, but they also mean that existing compliance requirements must be adapted to the changed regulations.

For illustration, Fig. 1 exemplifies regulatory changes using excerpts from different revisions of *AIFMD*, the European Union (EU) Directive on alternative investment fund managers (AIFMs) [10]. The example shows two non-consecutive legal provisions (at timestamp t_0=*08/06/2011*), namely Article 15(5) (labeled as p_1) and Article 33(6) (labeled as p_2). Tracing the changes over the course of AIFMD's lifetime, we see that a new subparagraph (labeled s_2) has been *added* to Article 15(5) at timestamp t_1=*20/06/2013*, leading to s_2 (in blue). Similarly, some text in Article 33(6) has been *replaced* at timestamp t_2=*01/08/2019*. Regulatory changes can also include the *deletion* of text or repeal of provisions, which we do not show in the figure.

Identifying such regulatory changes and understanding the impact of a change on existing compliance requirements is the first step for ensuring that the IT systems deployed by

[§] Equal Contribution

[†] Part of this work was done while the author was affiliated with SnT, University of Luxembourg.



At timestamp t_0 = *08/06/2011*

Article 15(5): The Commission shall adopt, by means of delegated acts in accordance with Article 56 and subject to the conditions of Articles 57 and 58, measures specifying: (a) the risk management systems to be employed by AIFMs in relation to the risks which they incur on behalf of the AIFs that they manage; […].

At t_1 = *20/06/2013*

Change described in *Article 3 Par 1 point 3 of directive 2013/14/EU*

"in paragraph 5, the following subparagraph is *added*: { s_2 }"

Article 15(5): The Commission shall adopt, by means of delegated acts in accordance with Article 56 and subject to the conditions of Articles 57 and 58, measures specifying: (a) the risk management systems to be employed by AIFMs in relation to the risks which they incur on behalf of the AIFs that they manage; […].

The measures specifying the risk-management systems shall ensure that the AIFMs are prevented from relying solely or mechanistically on credit ratings for assessing the creditworthiness of the AIFs' assets.

Article 33(6): If, pursuant to a planned change, the AIFM's management of the AIF would no longer comply with this Directive […], the competent authorities of the home Member State of the AIFM shall inform the AIFM without undue delay that it is not to implement the change.

At t_2 = *01/08/2019*

Change described in *Article 2 par 1 point 5 of directive 2019/1160/EU*

"in Article 33(6), the second subparagraph is replaced by the following: {s_3}"

Article 33(6): If, pursuant to a planned change, […] the relevant competent authorities of the home Member State of the AIFM shall inform the AIFM within 15 working days of receipt of all the information referred to in the first subparagraph […].

Fig. 1: Example of regulatory changes: s_2 has been added to p_1 and the first subparagraph in p_2 has been replaced by s_3

various stakeholders remain compliant after the change occurs. Once a change is identified, it can be traced back to the existing requirements by performing requirements traceability, and then the change impact of legal requirements can be analyzed to deduce recommendations on how to adapt existing software requirements to keep them compliant. Requirements change impact analysis can also be performed to understand how changing an existing requirement would affect other interdependent requirements. We note that the work presented in this paper is a piece of a larger research agenda [11].

As part of maintaining compliance requirements, requirements engineers must account for regulatory changes by adapting existing requirements or further specifying new requirements to ensure the regulatory compliance of software systems. For instance, the addition of s_2 in the figure requires specifying new requirements for implementing the new risk management practice in a lawful way. Similarly, the replacement of the phrase "*without undue delay*" in p_2 with "*within 15 working days*" (s_3 in the figure) impacts the time constraint for communicating the procedure of handling a planned change. This simple change may entail adapting existing communication methods to take into account the new time constraint.

Identifying changes in a text can be done with a simple *TextDiff* method. However, understanding that the changed text, e.g., in s_3, describes a time constraint, which can possibly impact the communication process, requires in-depth semantic analysis that is beyond *TextDiff* capabilities. Changes in regulations are often described in a dedicated legislative document (see the changes descriptions in Fig. 1) without necessarily providing the text of the modified legal provisions (i.e., s_1–s_3 in the figure). In such cases, even detecting the textual changes in regulations goes beyond *TextDiff*. Analyzing changes when the modified provisions are not given alongside the original ones is left for future work.

Regulatory compliance is a prominent research strand in RE. Various approaches have been proposed for extracting semantic information [12] or rights and obligations from regulations [13], [14], for modeling requirements variability [15], and for reconciling requirements from multiple jurisdictions [16]. Other approaches involve formalizing legal provisions through conceptual modeling [17], [18], [19], [20] or (semi-)formal specifications [21], [22]. Gordon and Breaux [23] propose an approach for manually specifying legal requirements in semi-formal representation that is then translated to logical expressions, for identifying, upon a change in a product requirement, IT system, or regulatory context, the set of legal requirements that the system must fulfill. Existing work in RE has two main limitations. First, extracting semantic information is mostly done through manually-defined rules that utilize traditional natural languages processing (NLP) methods of syntax parsing. Since then, the NLP landscape has seen a drastic change with the rapid adoption of large language models (LLMs) for solving various NLP downstream tasks more effectively [24], [25]. Second, approaches investigating compliance requirements mostly rely on manual formalization methods and do not address regulatory changes over time. To address these limitations, we propose an automated approach, namely MURCIA (MUlti-layered Regulatory Change Identification and Analysis), that leverages LLMs to automatically identify and analyze the regulatory changes over time.

Contributions. The contributions of the paper are as follows:

(1) We propose a taxonomy that characterizes regulatory changes at different levels of abstractions, organized in four layers. The *textual layer* describes a change in text at different levels of granularity, e.g., the replacement of a phrase or the addition of a sentence. The *semantic layer* captures the concepts that have changed in the terms and locutions, e.g., the addition of a *reference* (i.e., an identifier of an existing legal act). The *deontic layer* looks at the legal interpretation of the provision, e.g., the deletion of an applicability condition

with respect to a given *addressee* (i.e., the agent to whom the change is relevant). Finally, the *pragmatic layer* describes the change considering a concrete application context, both from RE and legal perspectives. The main objective of the taxonomy is to facilitate legal requirements analysis for the purpose of demonstrating regulatory compliance. We elaborate our taxonomy in Section III.

(2) We have built, as part of our work, a dataset of regulatory changes covering four widely used regulations from the finance domain. These regulations went through changes over time resulting in a total of 25 revisions that collectively contain 1293 provisions affected by at least one change at some point in time. The dataset has been manually curated by a third-party legal expert following our proposed taxonomy.

(3) We devise MURCIA to provide automated support for analyzing regulatory changes. Given two consecutive versions of a legal document as input, MURCIA builds on NLP technologies — in particular, the recent generative LLMs such as GPT [26] — to identify the regulatory changes according to our taxonomy. We empirically evaluate MURCIA on the dataset created in (2). Our results indicate that MURCIA can accurately identify textual changes with a precision of 87.8% and a recall of 93.5%. MURCIA can provide the meanings of the legal provisions (i.e., at the semantic layer) with a precision of 88.8% and recall of 92.9%, and further interpret the provisions (i.e., at the deontic layer) with a precision of 77.5% and recall of 91.1%.

Data Availability. To foster future research, we release both our dataset and evaluation material in an online annex [27].

Structure. Section II provides some background information. Section III introduces our taxonomy of regulatory changes. Section IV describes the MURCIA approach. In Section V, we report on the empirical evaluation, including prompt design. Section VI positions our work against the related literature. Section VII concludes the paper.

II. Background

Large language models (LLMs). Language Modeling is a traditional task in NLP which is concerned with determining the probability of the next word in a given text sequence [28]. Over a short period of time, LLMs have come to dominate the NLP state-of-the-art and demonstrated effectiveness in addressing challenging tasks without being explicitly trained to do so [29], [26], [30], [31]. LLMs are huge computational models with trillions of parameters, pre-trained on a massive amount of unlabeled corpora. Early LMs, e.g., BERT [32], are based on the Transformer architecture [33] with self-attention mechanisms that enable the model to capture the long-range dependencies and semantic interrelations in text. Fine-tuning LLMs with reinforcement learning and human feedback (RLHF) has led to the recent breakthrough in LLMs, featured by ChatGPT [34]. RLHF allows the model to generate responses that are closer to those of humans. With the growing size of LLMs and their powerful capabilities, classical fine-tuning (i.e., adjusting the parameters of a pre-trained model for a specific dataset and task) is no longer feasible without having dedicated, powerful resources. Alternatively, one can use prompt engineering which we explain next.

Prompt Engineering. Fine-tuning of LLMs has taken the form of designing prompts that provide the LLM with detailed and clear instructions about a target task and desired output. This process is referred to as *prompt engineering*. The NLP literature reports on various prompting strategies [35]. Below, we briefly discuss the ones we experiment with in this paper.

Zero-shot (ZS) prompting [26], [30] is a standard strategy formulating the prompt as an instruction about a specific task, e.g., the ZS prompt "What is the sentiment in the following product review?" aims to solve the sentiment analysis task, while "Summarize the following paragraph" aims to solve the text summarization task.

Few-shot (FS) prompting [26] is a strategy that provides, in addition to the instruction, a set of examples containing the input and output for a given task. This strategy assumes that a "few" labeled examples are available instead of large labeled datasets that were previously prepared and used for training a model from scratch, e.g., using machine learning (ML), or fine-tuning the parameters of an LLM for a specific task. The following example prompt applies the FS strategy: "Translate from English to French: red wine ⇒ vin rouge, cheese ⇒ fromage, newspaper ⇒ journal".

Chain-of-Thought (CoT) prompting [31] is a strategy which relies on the reasoning capabilities of the LLM. Recent studies [34], [36] suggest that appending the sentence "Let's think step by step" to the original prompt activates the LLM's reasoning capabilities and enables more accurate answers [36].

III. The Taxonomy of Regulatory Changes and their Implications on Compliance

In this section, we introduce our taxonomy and explain our methodology for building it.

Representing regulatory change. When a legislative act modifies another legislative act, the modification is introduced in terms of textual changes (for example, the provision: "point (ii) in Article 4(1) is deleted"). The change in the text affects both the expressed legal statements (i.e., the meaning directly conveyed by the text, taken in isolation) and the legal norms (i.e., the rule(s) resulting from the interpretation of the text as applicable to a specific addressee), however the latter cannot be inferred only from the textual change. For example, the addition of new text can entail introducing an exception, thereby reducing the area of application of the regulation itself. This poses the challenge of describing the impact of a textual modification not only on the regulatory text, but also on the norms expressed therein.

In this paper, we propose a taxonomy aiming at representing the different types of impact that a regulatory change may have on the compliance process.

Methodology. To build the taxonomy we followed and adapted a well-established methodology to build semantic resources, based on competency questions [37]. We started from the following questions that an expert would ask when dealing with a regulatory change: *(1) What has changed in the text? (2) What has changed in the legal statement? (3) What has changed in the legal norms in relation to a type of addressees?*

We identified three layers, one for each question, namely: *textual layer*, *semantic layer* and *deontic layer*. Note that none of these layers answers another important question regarding compliance in the legal practice, i.e., "what is the actual meaning of the change for the specific case of a given IT system?", which is the question legal experts are concerned with when checking the compliance of a specific IT system against the applicable law. This is beyond the RE perspective and is out of the scope of this paper.

We then looked at occurrences of regulatory changes on two sample EU legislative acts, namely AIFMD (introduced in Section I) and AIFMR [38]. For each observed change, we answered the aforementioned questions by identifying and organizing the relevant concepts in a taxonomy, following the theory of logical formulation of norms [39] in legal informatics [40], [41]. We complemented our taxonomy with elements from the literature [42], [13], [43], [12] to ensure its completeness beyond elements that were found in the sample regulations.

The Taxonomy. Table I illustrates the taxonomy, organized in three layers and elaborated below. For each layer, the table lists the categorization of the change according to our taxonomy. For instance, the taxonomy categorizes the textual changes at the textual layer (① in the table) into three levels of granularity, namely Paragraph, Sentence, and Phrase. The semantic and deontic layers (② and ③ in the table) are further categorized into sub-layers describing the changes both at concept and statement levels. For each category, the table further provides a description (**D**), lists the possible change types (**C**), and gives an example (**E**).

(1) Textual layer: This layer describes the textual modification. The biggest possible unit is the legal paragraph, i.e., the subdivision of the article into text paragraphs (which are always numbered in legislative acts). We have identified three levels of granularity to capture textual impact: *paragraph* (intended as the subdivision of an article of law), *sentence* (the text span delimited by a sentence ending indicator, i.e., fullstop [14]), and *phrase* (a text segment in a sentence). For all granularity levels, the impact is qualified as *addition*, *replacement* or *repeal* (i.e., deletion).

(2) Semantic layer: In this layer, we distinguish the impact on the statement as a whole and the impact on the concepts expressed therein. For statements, this layer represents the modification to the legal statement as expressed by the single textual provision, considering only the meaning that is directly conveyed by the text. For example, the subparagraph added in p_1 (s_2 in Fig. 1) is only seen as an obligation for the subject of the sentence (i.e., the "Commission") and not for entities playing other roles (such as the "AIFMs"). In this layer, the possible labels for statements are *regulative statement* and *constitutive statement*, further specified into *addition*, *replacement*, *repeal*. Both statement types are important for regulatory compliance. While regulative statements directly introduce requirements, constitutive statements include definitions and rules regarding the validity and efficacy of legal acts [40].

At the level of the individual concepts, this layer describes the change in the entities that are expressed by the text, according to a simple model derived from the literature in legal informatics [42] and RE [13], [12], with a notable difference for *event*. While the RE literature defines "action" as the main verb in the legal rule, we instead use the term "event" and extend the definition to include any action or event described in the text, and use the element "required action" in the deontic layer to represent the activity affected by a legal rule (norm).

(3) Deontic layer: In this layer we evaluate the impact of the change across the deontic space (i.e., the space of what ought to be done). We introduced this layer, because, from a legal perspective: (1) evaluating a statement in isolation (as in the semantic layer) is different from evaluating it together with related statements, and (2) interpreting the norms expressed in a statement highly depends on which addressee is considered. For example, interpreting s_2 (in Fig. 1) from the point of the Commission is different from evaluating it from the point of view another possible addressee, e.g., an AIFM. Specifically, the *constraint* for the Commission ("relying solely or mechanically") is instead a *required action* for an AIFM. The deontic layer thus involves the *interpretation* of the norms expressed by legal provisions. For this reason, before performing the analysis of the deontic impact, it is necessary to resolve all indirect obligations (i.e., specify the addressee from which point of view we will represent the norm) as well as all references (so that no aspect of the required action remains implicit). We note that in this layer multiple representations are possible for the same regulatory change.

At a statement level, it is possible to specify whether the impacted norm is a *constitutive* or a *regulative* one. The possible specific change values are *restricted*, *undefined*, and *permissive*. The focus of the regulatory change here is on the effort required from the addressee to obtain compliance. The change is restrictive if the effort is increased, e.g., a permission is now conditioned or an exception to an obligation has been removed. Conversely, the change is permissive if the effort is decreased, e.g., an obligation is now subject to a new precondition. Where the required effort is not comparable, the change is labeled as undefined. At the level of the individual concepts, we revert to assessing *addition*, *replacement* and *removal*, but we now target roles in the norm rather than concept types as we did in the semantic layer. The list of deontic concepts is derived from the literature [12], [43], [13]. Fig. 2 shows the semantic and deontic concepts in s_2 for the example provision p_1 depicted in Fig. 1.

Limitations. We note that we created our taxonomy of regulatory changes following a theory of legal norms representation which, despite being based on the literature [40], [41], is not univocal. For example, legal permissions can be formalized as *strong permissions* or *weak permissions* [44]. Different formalizations affect the granularity of the annotations. A second

TABLE I: Taxonomy with three layers (①: Textual, ②: Semantic, ③: Deontic). Legend: Description (**D**), Change Type (**C**), Example (**E**).

① Paragraph— **D:** A subdivision of an article of legislation, usually qualified by a cardinal number. **C:** *Addition*, *Replacement*, *Deletion*.
Sentence—**D:** The text span delimited by a typical sentence ending. **C:** *Addition*, *Replacement*, *Deletion*.
Phrase—**D:** A text span that is shorter than a sentence. **C:** *Addition*, *Replacement*, *Deletion*.

② **Statement:** Regulative Statement—**D:** Statement which, when considered in isolation, expresses some regulative norms. **C:** *Addition*, *Replacement*, *Deletion*. Constitutive Statement—**D:** Statement which, when considered in isolation, expresses no regulative norms. **C:** *Addition*, *Replacement*, *Deletion*.

Concept: Person—**D:** An individual; it may be a natural (or physical) person, or a juridical (or legal) person [42]. **E:** "financial institution". **C:** *Addition*, *Replacement*, *Deletion*.
Artifact—**D:** A human-made object (physical or virtual). **E:** "the agreement" [12], [42]. **C:** *Addition*, *Replacement*, *Deletion*.
Event—**D:** An action performed by an agent (e.g., publish) or a state of affairs. **E:** "to include", "available". **C:** *Addition*, *Replacement*, *Deletion*.
Reference—**D:** An identifier of a legal act. It can identify the entire act or a structural element within it [12]. **E:** "previous paragraph", "Article 13 of Directive 65/2009". **C:** *Addition*, *Replacement*, *Deletion*.
Time—**D:** A term or clause expressing a temporal constraint. It can be a date, a time span (two weeks, less than two weeks), or a duration (within reasonable time, as soon as possible) [12]. **E:** "two weeks", "within reasonable time". **C:** *Addition*, *Replacement*, *Deletion*.
Location—**D:** A term or clause describing a (physical or virtual) location [12]. **E:** EU. **C:** *Addition*, *Replacement*, *Deletion*.

③ **Statement:** Regulative Norm—**D:** Norms that prescribe obligations, prohibitions and permissions [40]. **E:** "the buyer must pay the price of the goods to the seller". **C:** *Permissive*, *Restrictive*, *Undefined*. Constitutive Norm—**D:** Norms that regulate the creation of institutional facts as well as the modification of the normative system itself [40], [41]. **E:** "the price of the goods is intended as the price advertised or agreed". **C:** *Permissive*, *Restrictive*, *Undefined*.

Concept: Addressee—**D:** The actor performing the required action [12], [13]. **E:** in "the buyer must pay 100 dollars", the addressee is *the buyer*. **C:** *Addition*, *Replacement*, *Deletion*.
Beneficiary—**D:** An intermediary or beneficiary of the required action [12], [13]; Beneficiaries are often expressed as indirect objects. **E:** In the obligation "to pay 100 dollars to the seller" the beneficiary is *the seller*). **C:** *Addition*, *Replacement*, *Deletion*.
Target—**D:** An entity affected by the required action [12], [13]; Here, target is intended as target of the required action (that which is normally expressed by a direct object). **E:** In "to pay 100 dollars" the target is *100 dollars*. **C:** *Addition*, *Replacement*, *Deletion*.
Required Action—**D:** An event that is to be performed by the addressee and that is deontically qualified (i.e., required, prohibited, or allowed) by the norm [43]. **E:** in "the buyer must pay 100 dollars", the required action is *to pay*. **C:** *Addition*, *Replacement*, *Deletion*.
Pre-condition—**D:** A circumstance whose verification triggers the application of the norm [43]. **E:** in the sentence "if the buyer has paid the price, the seller must deliver the goods", *if the buyer has paid the price* is a pre-condition for the seller to have the obligation to deliver the goods. **C:** *Addition*, *Replacement*, *Deletion*.
Constraint—**D:** A circumstance whose verification is necessary for the required action to be fulfilled. **E:** In the obligation "to pay within 5 days", *within 5 days* is the constraint [43]. **C:** *Addition*, *Replacement*, *Deletion*.

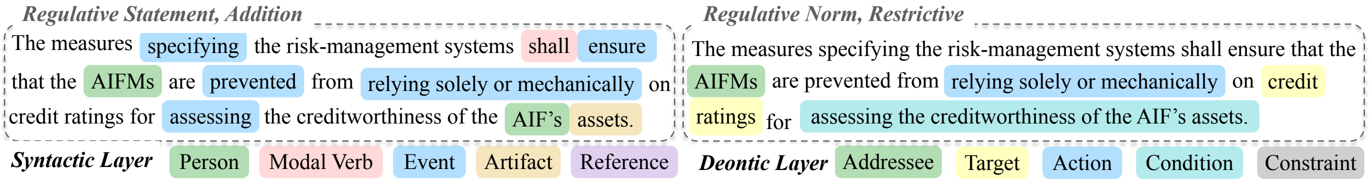

Regulative Statement, Addition
The measures specifying the risk-management systems shall ensure that the AIFMs are prevented from relying solely or mechanically on credit ratings for assessing the creditworthiness of the AIF's assets.
Syntactic Layer
Person
Modal Verb
Event
Artifact
Reference
Regulative Norm, Restrictive
The measures specifying the risk-management systems shall ensure that the AIFMs are prevented from relying solely or mechanically on credit ratings for assessing the creditworthiness of the AIF's assets.
Deontic Layer
Addressee
Target
Action
Condition
Constraint

Fig. 2: Example of semantic annotations (left side) and deontic annotations (right side) in s_2 of the example p_1 in Fig. 1.

limitation is that classifying the changes (particularly at the deontic layer) may vary in certain cases based on the way the elements constituting the norm are interpreted and combined, especially when considering implicit addressees. However, we decided to keep the taxonomy generic enough to be more understandable to legal experts, who do not handle well highly formalized languages. To assess the practical applicability of our taxonomy, we conducted validation sessions with three legal experts, namely a senior lawyer with more than 10 years experience in banking and financial Law — with an emphasis on the corporate governance of banks and financial institutions — and two law students. Their feedback confirmed that most of the concepts in our taxonomy are straight-forward while other concepts (e.g., constraint) might require guidelines about their usage, especially in complex situations.

IV. Automated Regulatory Change Identification

In this section, we describe our automated approach for identifying the regulatory changes, MURCIA. We envision MURCIA as an automated assistant to a human analyst instead of being a fully automated approach. The rationale behind this vision is the following: (i) The legal domain is complex and requires domain expertise for interpretation; (ii) despite having powerful capabilities, current technologies often require manually crafted, precise instructions that can be optimized in iterative interactions with the human analyst.

Fig. 3 shows an overview of MURCIA. MURCIA takes as input two legislative acts, a base act (B) and a modified act (M). In step 1, MURCIA pre-processes B and M and divides them into provisions. Steps 2, 3, and 4 involve identifying the regulatory changes, as per our taxonomy (see Section III), at the textual, semantic, and deontic layers, respectively. The regulatory change analysis in each step is done by prompting a generative LLM (such as GPT). We explain our prompt engineering strategy in Section V-C. Finally, MURCIA combines the intermediary outputs generated in steps 2–4 and returns the list of regulatory changes. Below, we elaborate each step.

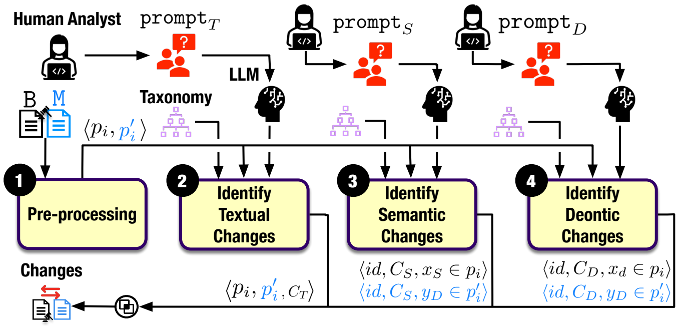

Human Analyst
$\texttt{prompt}_T$
$\texttt{prompt}_S$
$\texttt{prompt}_D$
LLM
Taxonomy
B M
$\langle p_i, p'_i \rangle$
1 Pre-processing
2 Identify Textual Changes
3 Identify Semantic Changes
4 Identify Deontic Changes
Changes
$\langle p_i, p'_i, C_T \rangle$
$\langle id, C_S, x_S \in p_i \rangle$
$\langle id, C_S, y_D \in p'_i \rangle$
$\langle id, C_D, x_d \in p_i \rangle$
$\langle id, C_D, y_D \in p'_i \rangle$

Fig. 3: Overview of MURCIA (B: base act, M: modified act, p_i and p'_i: textual strings of provisions in B and M, respectively. C_T, C_S, and C_D: textual, semantic, and deontic changes identified between p_i and p'_i, respectively. $\texttt{prompt}_T$, $\texttt{prompt}_S$, and $\texttt{prompt}_D$: prompts to identify textual, semantic, and deontic changes, respectively)

Step 1: Pre-processing. This step takes as input a base act B which is applicable at time t_{B} and a modified act M, that includes the regulatory changes introduced in B and is applicable at time t_{M}. Step 1 starts with the assumption that M is available. We note that M is not always available; however, developing an automated approach for creating M, though beneficial, is out of the scope of this work.

MURCIA generates two lists from the input acts, namely B= $[p_1, \ldots, p_n]$ and M= $[p'_1, \ldots, p'_n]$, where p_i and p'_i are textual strings of the provisions in B and M, respectively. To generate these lists, we apply a simple NLP pipeline, composed of *tokenization* and *sentence splitting*. Using this pipeline, MURCIA breaks the running text in B and M into a list of paragraphs, demarcated via recognizing carriage returns. These paragraphs are further divided into sentences. The textual representation in our work (p_i or p'_i) is generated at the sentence level. Our rationale for focusing on this granularity level is that changes occurring at other granularity levels can be captured through changes in sentences. Changes in a paragraph correspond to the combination of the changes in the sentences therein, whereas changes at the phrase level are still captured in our sentence-level analysis.

In this step, MURCIA further uses a simple text *TextDiff* function to create pairs of provisions in order to identifiy the revised provisions and characterize the change. Given the two lists of provisions from B and M, MURCIA maps the provisions in B to their corresponding provisions in M and return, using the *TextDiff* function, for each pair of provisions $\langle p_i, p'_i \rangle$, one of four values: (i) *equal* when p_i in B is found as-is in M. This value indicates no change and is thus disregarded in our work. (ii) *modified* when p_i in B has been modified in M. This value indicates the change type *replacement* and is represented as $\langle p_i, p'_i \rangle$, where p_i and p'_i are both not empty and $p_i \cap p'_i \neq \phi$. (iii) *deleted* when p_i in B is not found in M. This value indicates *deletion* where p'_i is empty ($p'_i = \epsilon$). Finally, (iv) *added* when p'_i in M is not found in B. This value indicates *addition*, where $p_i = \epsilon$.

The pairs of provisions $\langle p_i, p'_i \rangle$ affected by any regulatory change are then passed on to steps 2–4 and are provided as inputs to the LLM to analyze the regulatory changes therein. While providing long text blocks to LLMs would be nowadays possible, the goal of our splitting strategy is two-fold: (i) We reduce the potential noise text in the legal act (e.g., preambles or footnotes) which is not relevant to regulatory change analysis; (ii) we build a more structured material that is easier to validate by the human analyst without necessarily navigating through the original acts.

Step 2: Identify textual changes. For each pair $\langle p_i, p'_i \rangle$ from step 1, MURCIA identifies the textual changes between p_i and p'_i at different granularity levels, namely sentence and phrase levels. Paragraph level changes are skipped as discussed above. Specifically, MURCIA applies a customized prompt (denoted as $\texttt{prompt}_T$) which instructs the LLM to generate as output all the textual changes C_T between p_i and p'_i at sentence and phrase levels. C_T describes the exact text that has changed as well as the corresponding change type *addition*, *deletion*, or *replacement*. We define the exact text of $\texttt{prompt}_T$ in Section V-C. The intermediary output of step 2 is the list O_T of triples $\langle p_i, p'_i, C_T \rangle$ describing the textual changes between between p_i and p'_i.

Step 3: Identify semantic changes. In this step, MURCIA analyzes the semantic changes between the input pairs p_i and p'_i. Similar to step 2, MURCIA enables the human analyst to instruct the LLM through a customized prompt (denoted as $\texttt{prompt}_S$) to identify the semantic changes. The exact text of $\texttt{prompt}_S$ is provided in Section V-C. Given the textual changes identified in the previous step, $\texttt{prompt}_S$ focuses on identifying the different semantic concepts in p_i and p'_i according to our taxonomy. For each concept outlined in Table I, step 3 generates as intermediary output the list O_S of triples consisting of: *id* (a unique identifier prefixing each provision in the input pair,) the *semantic concept* C_S from our taxonomy, and the actual *text segment* labeled with that concept ($x_S \in p_i$ and $y_S \in p'_i$).

Step 4: Identify deontic changes. In this step, we perform a similar analysis to that of step 3, but on the deontic layer. MURCIA enables prompting the LLM through a customized prompt (denoted as $\texttt{prompt}_D$) to analyze the deontic concepts in the input provisions. We present $\texttt{prompt}_D$ in Section V-C. The intermediary output of step 4 is the list O_D of triples consisting of: provision *id* as defined above, *deontic concept* (C_D), and the actual *text segment* labeled with that concept ($x_D \in p_i$ and $y_D \in p'_i$).

The final output of MURCIA is the triple $\langle O_T, O_S, O_D \rangle$, combining the intermediary outputs produced in steps 2–4.

V. Empirical Evaluation

In this section, we report on the empirical evaluation of MURCIA, which we performed by answering the following research questions (RQs):

RQ1: How accurate is MURCIA in identifying the textual changes in legal provisions and with which prompting

strategy can such an accuracy level be achieved? Identifying the textual changes between legal provisions according to our taxonomy (Step 2 of MURCIA) can be done through different prompting strategies. In RQ1, we identify the alternative prompting strategy that yields the best accuracy.

RQ2: How accurate is MURCIA in providing meanings and interpretations of legal provisions and with which prompting strategy can such an accuracy level be achieved? MURCIA identifies the regulatory changes at the semantic and deontic layers (Steps 3 and 4) by interpreting the legal text, i.e., identifying the concepts according to our taxonomy. This interpretation can also be obtained through different prompting strategies. In RQ2, we identify the strategy that enables MURCIA to produce the most accurate meanings and interpretations of the legal provisions.

A. Implementation

We have implemented MURCIA in Python 3.8 and Jupyter Notebooks [45]. Specifically, we implemented the NLP pipeline (step 1 in Fig. 3) using NLTK 3.5 [46] and difflib [47]. For steps 2–4, we queried the GPT models [26] through the OpenAI API, and for prompt engineering (in Section V-C), we used the web interface of ChatGPT (https://chat.openai.com).

B. Data Collection Procedure

Our data collection aimed to manually analyze regulatory changes according to our taxonomy. Specifically, we collected our data from several European directives regulating the financial market, described in Table II. The rationale behind selecting these regulations is two-fold. First, they have significant impact on the compliance of financial actors in Europe, e.g., fund management companies. Second, they were subject to multiple changes over their lifetime.

Our data collection was performed in two phases. The first phase took place during the taxonomy building where an expert with legal informatics background (the second author) analyzed AIFMR and AIFMD, as explained in Section III. In the second phase, a third-party annotator (a Law student to whom we refer with the pseudonym Jo) was hired to analyze all regulations in Table II, including the ones analyzed in the first phase. Jo has previous experience with annotating legal text for developing automated solutions that address regulatory challenges. Prior to starting his work, Jo underwent a training session where the taxonomy was first extensively introduced alongside examples. Jo was then instructed to annotate a small subset (equivalent to 10 provisions) according to the taxonomy and we had a feedback session to discuss borderline cases. To mitigate fatigue, we provided Jo with the revisions of the regulations in several batches over three months, on which he spent a total of 78 hours. Jo was further encouraged to limit his working periods to two hours at a time.

TABLE II: Statistics for our *Dataset*.

Act	Revisions	$\#\langle p_i, p_i\prime \rangle$	*Addition*	*Replacement*	*Deletion*
AIFMD	6	35	29	10	2
AIFMR	3	16	12	4	0
MIFID II	10	607	16	443	150
MIFIR	6	635	150	483	2

[§] *The acts are available through the EurLex platform under unique document IDs: 32011L0061 for AIFMD, 32013R0231 for AIFMR, 32014L0065 for MIFID, 32014R0600 for MIFIR.*

We shared with Jo the taxonomy, the original regulations available on EurLex platform, and a predefined *addressee* for each regulation (necessary for deontic interpretations). To help the analyst perform his manual task more efficiently, we also provided him with the automatically generated lists of pairs of provisions $\langle p_i, p_i' \rangle$ generated by MURCIA, alongside automatically derived textual change types (see step 1 in Section IV). Jo was instructed to examine each provisions pair, validate the textual change type, and provide the regulatory changes at the semantic and deontic level according to our taxonomy. We limited the annotation task to describing the nature of the change regarding a concept in the input pairs since demarcating the exact text of that concept would require tremendous time and effort. More precisely, if the analyst identifies at least one semantic or deontic concept in the input pairs, then he assigns a label to that concept indicating whether the concept is added, replaced, or deleted. The result of this manual analysis is our ground truth.

We measured the inter-rater agreement using Cohen's Kappa (κ) [48] on AIFMD and AIFMR. We obtained an average κ of 0.77 and 0.68 on the analysis of the semantic and deontic layers, respectively. Both values indicate substantial agreement. Computing κ values per semantic concept yielded an average ranging between 0.89 (almost perfect agreement) for identifying *location* and 0.6 (moderate agreement) for identifying *artifact*, whereas the average κ per deontic concept ranged between 0.82 (almost perfect agreement) for identifying the required *action* and 0.44 (moderate agreement) for *constraint*. It is clear that the deontic layer shows less agreement since the analysis is more sensitive to legal interpretation as discussed in Section III. However, we believe that these agreement values are sufficient in the legal domain [20] and for our analysis. We discussed the disagreements with Jo and (i) agreed on a shared understanding of the concepts which was consistently applied in the annotation, and (ii) improved the definitions and exemplification of these concepts in our annotation guidelines.

C. Prompt Engineering

In this section, we explain our strategy for designing the prompts prompt$_T$, prompt$_S$, and prompt$_D$ (in Fig. 3). The resulting prompts are then used for answering our RQs.

We experiment the three alternative prompting strategies, namely ZS, FS, and CoT we explained in Section II. To draw meaningful conclusions about the prompting strategies, we design and validate our prompts exclusively using ChatGPT (GPT3.5).

We design our prompts using AIFMR following an iterative process, outlined next.

Iteration 1: Observing the performance of ChatGPT. In the first iteration, we drafted several prompts variants aiming to assess the alternative prompting strategies through answering three main questions: Do prompt variants significantly affect the output of a ZS prompt? How does varying the examples, their number, and order affect the output of an FS prompt? Is the reasoning triggered by CoT and provided alongside the output meaningful? Inspired by the work of Yu et al. [36], we focused on varying the text of the prompt to examine the effect on the ZS prompt, whereas we focused on varying the examples provided to the LLM to assess the FS prompt. We then coupled these variants with the reasoning statement (explained in Section II) to assess the COT prompt. This procedure resulted in 15 variants of prompting text for prompt$_T$, 12 of prompt$_S$, and 19 of prompt$_D$. We note that we opted to drop some of the variants in the case of prompt$_S$ (e.g., varying the text of the prompt for ZS) for two reasons. First, these variants are very similar (e.g., "determine the regulatory changes" versus "determine the changes") and would be already assessed in prompt$_T$. Second, running the prompts was often interrupted and delayed due to the traffic on the OpenAI servers. Prompt$_D$, on the other hand, targets the deontic layer of our taxonomy which is more complex as it involves interpreting the legal text. Thus, we investigated more variants to assess the capability of the LLM in providing plausible interpretations, e.g., for a given *addressee*.

We then used the prompts variants of all prompting strategies and instructed the GPT model to identify a total of 16 changes introduced in AIFMR over its lifetime. For the FS prompts, we randomly selected from AIFMD examples that capture the three change types (addition, replacement, and deletion). Finally, we manually validated the output (including also the rationale provided by the model) with respect to the type of legal analysis expected in the three layers of our taxonomy. An output is labeled as *correct* if it contains all needed information for the regulatory change analysis, *partially correct* if it contains only a subset of the information, *incorrect* if it contains wrong information, and *irrelevant* if it does not contain information related to the regulatory changes. We carefully analyzed the results and discussed our observations, outlined below. For space limitations, we provide the prompts and our validation results in the online annex [27].

O1: While ZS is not sensitive to minor variations, it is highly sensitive to the details regarding our taxonomy of regulatory change

O2: ChatGPT provides inconsistent results. Forcing an output format (e.g., JSON) reduces this effect drastically since the model will have to generate content according to the desired task.

O3: The number of examples plays a significant role in educating ChatGPT about how the task should be solved, whereas which exact examples and in which order they are provided have less impact on the performance of the model in the case prompt$_T$. The same observation does not hold for prompt$_S$ or prompt$_D$. We believe the reason is that the semantic and deontic layers require more in-depth interpretation that is not straightforward to convey through examples.

O4: While the explanation provided by ChatGPT in CoT are not always useful, ChatGPT's performance still improves when combining CoT with FS. This can be justified by the fact that activating the reasoning capabilities of the model is an important step for identifying changes more accurately.

Iteration 2: Refining the prompts. In this iteration, we refine the prompts to improve the performance of ChatGPT. To address our observations highlighted above, we edit the prompts as follows. First, we expose the task of regulatory change analysis right at the beginning of the prompt. Second, we always ask for a certain output format, which forces ChatGPT to follow the same terminology we have in our taxonomy (e.g., replacement instead of modification). Third, we clearly distinguish between examples and text-to-analyze instances. Following this, we iteratively refine the prompts and re-validate the provisions that were previously marked as incorrect. A simplified version of the final prompts used to answer the RQs in our evaluation are listed below. Due to space limitations, we provide only ZS. The complete list is available in our online annex [27].

- ***prompt$_T$:***

```
Analyze the textual changes between the legal
provisions at different levels of granularity:
"sentence" and "phrase". Describe the changes
exclusively using the change types "addition",
"replacement", and "deletion". In case of
"replacement", analyze the change at the
level of phrases and define the exact text
that has been replaced. List all the phrases
which are subjected to change in a JSON file,
with "change-type", "granularity-level",
"changed-text" as keys and "changed-text" is
further refined using "old" describing the old
text, "new" describing the new text, as keys.
The legal provisions to analyze are delimited
with triple backticks, and provided next. old
text: '''{p_i}''', new text: '''{p'_i}'''
```

- ***prompt$_S$:***

```
Analyze the semantics of the legal provisions
provided below, as follows. (1) Define the
noun, verb and prepositional phrases in
each provision. (2) Assign a label to each
phrase exclusively from the following labels:
Modality: is a verb expressing modality (e.g.
"shall", "shall not", "must", "is prohibited").
Person: is a natural (or physical) person, or
a juridical (or legal) person. An artifact
is a human-made object (e.g., "document",
"agreement"). An event is an action performed
by an agent (e.g. publish) or a state of affairs
(e.g. include). reference is an identifier of
an existing legal act (e.g., "Article 32", "the
previous paragraph"). time is a term or clause
expressing a temporal constraint (e.g., "as soon
as possible", "within one week", "19 december
2020"). location a term or clause describing
a (physical or virtual) location (e.g., "in
Europe", "through the local branch"). null if
none of the above is applicable. (3) Generate
the output as JSON format, using "provision-id",
"phrase" and "label" as keys. old-provision:
'''{p_i}''', new-provision: '''{p'_i}'''
```

- ***prompt$_D$:*** Exactly the same as prompt$_S$ except that we provide in instruction (2) the deontic concepts instead of the semantic ones:

TABLE III: Accuracy of MURCIA.

	ZS			FS			CoT+FS		
	P	R	F_1	P	R	F_1	P	R	F_1
	Textual Change Detection (**RQ1**)								
GPT3.5	45.5	74.1	56.3	53.2	71.7	61.1	66.0	75.6	70.5
GPT4	68.8	88.0	77.2	81.6	90.9	86.0	**87.8**	**93.5**	**90.5**
	Semantic Concepts Identification (**RQ2**)								
GPT3.5	90.5	67.8	77.5	86.2	77.6	81.7	87.1	77.8	82.2
GPT4	88.3	92.2	90.2	88.8	92.8	90.7	**88.8**	**92.9**	**90.8**
	Deontic Concepts Identification (**RQ2**)								
GPT3.5	79.5	60.0	68.4	78.2	78.5	78.3	61.0	75.7	67.5
GPT4	76.4	**91.2**	83.1	77.5	91.1	83.7	**86.4**	83.5	**84.9**

```
Addressee is an actor performing the required
action, required action is the main verb that
is deontically qualified by the norm, target
is the direct object of the required action
(e.g., the target is "100 dollars" in "to pay
100 dollars"), beneficiary is the indirect
object (e.g., the beneficiary is "the buyer" in
"pay 100 dollars to the buyer"), pre-condition
is a circumstance whose verification triggers
the application of the norm (e.g., "if the
buyer has paid the price"), constraint is a
circumstance whose verification is necessary for
achieving compliance of obligations or breach of
prohibitions (e.g., "pay within 30 days").
```

D. Accuracy of Textual Change Identification (RQ1).

Methodology. As discussed in Section V-B, our ground truth does not capture the exact text that has changed, rather the change type. However, during the taxonomy creation, we also marked the changed text segments but only for AIFMD (composed of 35 provisions pairs and 41 changes, see Table II). We thus answer RQ1 for AIFMD only. To do so, we prompted GPT3.5 and GPT4 models to identify the textual changes between legal provisions in AIFMD. To generate the results we applied $\texttt{prompt}_T$, which we refined in Section V-C according to three alternative prompting strategies, namely ZS, FS, and CoT. We define a *true positive (TP)* as a change that is correctly identified by the model, a *false positive (FP)* as the change that is falsely introduced by the model, and a *false negative (FN)* as the change that is missed by the model. Following this, we evaluate the results using *precision* (P), where $P = \frac{TPs}{TPs+FPs}$, *recall* (R), where $R = \frac{TPs}{TPs+FNs}$, and F_1 (the harmonic mean between precision and recall), defined as $F_1 = \frac{2*P*R}{P+R}$. We note that the changes captured by the model and our ground truth can be of different granularity. For instance, adding an itemized list of seven elements is counted as one change (*addition*) in our ground truth, but it is captured as seven additions by the GPT model. We considered such cases as correct. Therefore, we had to identify the correct and missing changes by carefully validating the results generated by the GPT models.

Results. The top part of table III lists the accuracy of MURCIA operationalized on GPT3.5 and GPT4 models, prompted using the three alternative prompting strategies presented above. The GPT4 model clearly outperforms GPT3.5 across all prompting strategies, with CoT+FS being the most promising strategy. Using CoT+FS, GPT4 yields a gain of 21.8 pp (pp: percentage points) in precision and 17.9 pp in recall. Despite providing context to the LLMs about the regulatory change task and further exposing our taxonomy in all prompts, using ZS in MURCIA does not yield good accuracy. The complexity of the task requires more sophisticated prompting strategies where the model is also shown few examples (FS) with an explicit activation of its reasoning capabilities (CoT). While we are mainly interested in assessing the accuracy of MURCIA for our application context, we note that using OpenAI GPT4 is, at the time of writing this article, is subject to a fee. There is thus a tradeoff between the accuracy of MURCIA and its operational cost.

From a practical standpoint, we favor recall over precision, since it is easier for a legal expert to review the identified changes than to spot missed changes in the legal text. In that regard, the results obtained when using GPT4 are promising.

The answer to RQ1: MURCIA can identify the textual changes with a precision of 87.8% and a recall of 93.5%, by prompting GPT4 using CoT+FS.

Error Analysis. We analyzed the causes of the errors made by MURCIA. The model generated a total of 49 changes, out of which six were falsely introduced changes. The model further misses three changes. These errors can be explained as follows.
(1) Granularity: The GPT model analyzes the provisions with a different granularity treatment as the one we have in our ground truth, e.g., a replacement of one sentence by two sentences can be captured as the replacement of one sentence and the addition of another. In our context, to obtain conservative results, we deemed such changes identified by the model as incorrect since different change types have different implications on software requirements (e.g., adding a sentence can introduce a new corresponding requirement while replacing the entire paragraph might require adapting existing requirements).
(2) Interpretation Errors: In some cases, the GPT model made mistakes in capturing the right change, e.g., falsely suggesting a replacement of a phrase when it was simply deleted.

E. Accuracy of Legal Interpretation (RQ2).

Methodology. To address RQ2, we prompted the GPT models to identify the semantic and deontic concepts in the legal provisions (i.e., we asked to interpret the legal text) in our dataset (see § V-B). More specifically, due to cost and traffic on the OpenAI API, we assessed the performance of the models on the entire AIFMD directive but only on subsets of provisions from MIFIR and MIFID II, consisting of 52 provision pairs containing a total of 560 changes and 65 pairs with 707 changes, respectively. We selected the provisions such that they contain, according to the ground truth, at least half of the concepts in our taxonomy (i.e., more than three semantic or deontic concepts). We also disregarded the pairs of provisions whose text (when combined with the prompt) exceeded 1000 tokens, to avoid hitting the threshold for maximum input token length of the GPT models. Breaking provisions into smaller units is not a plausible alternative in our context as the change will not be properly captured. We applied the refined prompts ($\texttt{prompt}_S$ and $\texttt{prompt}_D$) from Section V-C. To evaluate the results, we compare the semantic and deontic concepts identified by the GPT models against the ones we have in our ground truth. Following this, we define true positives (TPs) as the concepts that are identified by the GPT models and are further marked with any change type in our ground truth, false positives (FPs) as the concepts that are identified by the models but are not in our ground truth, false negatives (FNs) as the concepts that are introduced in our ground truth but are not identified by the models. We then report P, R, and F_1 as defined in RQ1.

Results. The bottom part of table III shows the results of GPT3.5 and GPT4 in identifying the semantic and deontic concepts. For the semantic layer, GPT4 outperforms GPT3.5 across all metrics and prompting strategies. The recall of GPT3.5 is not sufficient for solving RE tasks where recall is typically favored [49]. GPT4, however, provides high recall with acceptable precision, yielding a F_1 score over 90% for all three prompting strategies.

Results show a drop in the accuracy of the GPT models when it comes to the deontic layer, with GPT4 still faring better than GPT3.5. This drop is expected, considering the complexity of the required task, which involves legal interpretation and, in some cases, contextual information which the GPT model cannot retrieve by simple means. Unlike the semantic layer, using ZS generates the best recall of 91.2% with a precision of 76.4%. Using FS yields similar values for precision and recall. In comparison, using CoT+FS yields the best precision value with an average gain of ≈ 9 pp, but at the expense of an average loss of ≈ 8 pp in recall. This difference between FS and CoT+FS highlights the fundamental role of domain-specific knowledge. The reasoning of the model led to interpretations (that might be valid in a generic context) but are yet not in line with the ones we have in our ground truth. In contrast, identifying semantic concepts does not require in-depth interpretation.

Having high recall value is a good prospect, as it ensures that few changes are missed, while a low precision value can be addressed by a focused, manual review aimed at spotting the most common errors. In our context, we favor recall, and recommend FS as the prompting strategy for deontic concepts.

The answer to RQ2: Using GPT4, MURCIA can identify the semantic concepts with a precision of 88.8% and recall of 92.9% (adopting a CoT+FS prompt), and it can further identify the deontic concepts with a precision of 77.5% and recall of 91.1% (using an FS prompt). By averaging the results obtained for both semantic and deontic concepts, we can say MURCIA can provide the meaning and further interpret the legal text with a precision of 83.1% and recall of 92.0%.

Error Analysis. We observed error causes similar to those in RQ1. However, we also identified additional causes:

(1) Deontic interpretation with respect to an addressee. Our ground truth contains deontic interpretations with respect to a specific addressee. Our $\texttt{prompt}_D$, however, does not explicitly limit the analysis to an addressee for two reasons. First, our preliminary experiments showed that the GPT models get confused when a specific addressee is mentioned in the prompt. Second, we opted to keep the prompts generic to capture complete information at the deontic layer to better support regulatory change analysis.

(2) Implicit mentions of concepts. In some cases, the legal provisions implicitly refer to a concept that cannot be clearly identified in the text without domain knowledge, e.g., a beneficiary that is inferred knowing the regulation context.

F. Threats to Validity

Internal Validity. One of the main concerns of internal validity is bias. We mitigate bias by delegating the creation of our ground truth to a third-party annotator who had no exposure to the solution design. The two regulations that were analyzed during the creation of our taxonomy (namely AIFMR and AIFMD) were re-labeled by the third-party annotator before using them again for designing the prompts and answering RQ1 in our evaluation.

Construct Validity. The same textual changes can be identified in multiple ways depending on the granularity level, e.g., replacing a sentence (i.e., replacement) can be due to only adding a phrase therein. To account for such cases, our evaluation is performed at the change level regardless of granularity.

External Validity. To gain sufficient confidence in our observations, designing the prompts was done by one researcher (an author of this paper) while the results were validated by different researchers (another two authors). To prevent the LLMs from learning the data that we experiment with, the regulation that is used for designing the prompts is different from the ones used in evaluation. We further evaluated our approach on multiple regulations covering different regulatory changes. While our taxonomy is not specific to financial regulations, further experimentation on regulations from other domains would improve the generalizability of our results.

VI. Related Work

Our work is related to change analysis in RE and to analyzing edits in text revisions; the latter, in the context of NLP, is the task equivalent to change analysis.

Change analysis in RE. Analyzing the impact of changing requirements on other software artifacts is a long-standing problem in RE [50], [51], [52], [53]. There exists an extensive body of work on analyzing legal requirements for different purposes, which mostly focuses on legal requirements elicitation and formalization as well as semantic analysis [13], [21], [14], [54], [18], [20], [12], [1], [55]. However, little attention has been given to regulatory change analysis.

Maxwell et al. [56] analyzed changes in the law to select and prioritize legal requirements analysis through predicting the provisions in draft or future acts that are likely to evolve over time. Saito et al. [57], [58] investigated how to visualize the dependency between software requirements and other software artifacts, including legal requirements. Gordon and Breaux [59], [16] as well as Ben Nasr et al. [15] provided approaches for analyzing and comparing requirements from multiple jurisdictions. Breaux et al. [21] proposed a legal requirements specification language (LRSL) for representing legal requirements from multiple regulations, to facilitate traceability between technical requirements and legal provisions. Building on their previous work, Gordon and Breaux [23] proposed a framework for identifying legal requirements that are applicable for an IT system according to technical requirements and design specifications of that system. Their framework produces a coverage model which enables analyzing the changes in both the system features (and hence its requirements) as well as the changes in law.

None of the these works has addressed the identification of regulatory changes occurring over a period of time, while further capturing the nuances in the legal interpretation resulting from these changes. Furthermore, semantic information extraction in RE has been performed so far through manually-defined rules over syntax parsing. In contrast, our work presents a taxonomy for analyzing regulatory changes both at the textual level as well as at the semantic and deontic levels. Our work further proposes a semi-automated approach that leverages recent NLP technologies for assisting the human analysts in identifying and understanding the regulatory changes. Our approach is a first step towards addressing the change impact analysis of regulations on software requirements.

Text revisions in NLP. NLP has been widely applied in the legal domain [60], [61], [62]. This research targets various tasks such as text classification of cases [63], named entity recognition [64] and information extraction [65] from legal documents. Recent work has also investigated the performance of LLMs in solving different legal tasks [66], [67]. Our work on prompt engineering is inspired by this latter research.

Another strand of research focuses on analyzing revisions of text documents, e.g., resulting from editing Wikipedia articles. Daxenberger and Gurevych [68], [69] created a corpus showing the different categories of possible edits in text revisions, e.g., paraphrasing or relocating text. The authors further proposed a classification method to classify edit categories automatically. Yang et al. [70] presented a taxonomy of the intentions of Wikipedia edits; e.g., "elaboration" means that a new content is added or existing content is extended to improve its meaning. The authors further investigated automated means for identifying edits' intentions. More recently, Spangher et al. [71] proposed using LLMs for classifying text change, namely addition, deletion, editing, and refactoring (i.e., relocating text). Du et al. [72] presented a large-scale, multi-domain, edit-intention-annotated corpus capturing iterative text revisions and studying the evolution of text quality across iterations.

Compared with these approaches, our work specifically focuses on analyzing the changes in regulations throughout their lifetime. While it is sufficient for NLP to work on the text at the sentence level, our work not only analyzes the changes at phrasal level but captures (by dividing the text into phrases) how legal interpretation is affected by a change in a more comprehensive manner.

VII. Conclusion

In this paper, we have proposed MURCIA, a semi-automated approach for analyzing regulatory changes taking effect throughout the lifetime of regulations and thus supporting compliance analysis. To build MURCIA, we first created a taxonomy that characterizes and organizes the regulatory changes into four layers. Our automation then leverages GPT models for identifying the changes according to the taxonomy. We have evaluated MURCIA on regulations from the financial domain. Our results show that MURCIA can identify *textual changes* in provisions with a precision of 87.8% and recall of 93.5%. Moreover, MURCIA can provide the meanings and interpretations of legal provisions with an average precision and recall of 83.1% and 92.0%, respectively.

Our work provides a first significant step towards an end-to-end pipeline for change impact analysis of regulations on software requirements. In the future, we plan to investigate the impact of identified changes on existing requirements. To demonstrate how our work is used in practice, we also plan to investigate a concrete use case of a system that is affected by regulatory changes (pragmatic layer in our taxonomy).

Acknowledgment

This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant reference NCER22/IS/16570468/NCER-FT. For the purpose of open access, and in fulfillment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.
Lionel Briand was partially supported by the Science Foundation Ireland grant 13/RC/2094-2, and the Canada Research Chair and Discovery Grant programs of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] S. Abualhaija, C. Arora, A. Sleimi, and L. C. Briand, "Automated question answering for improved understanding of compliance requirements: A multi-document study," in *30th IEEE International Requirements Engineering Conference, RE 2022*. IEEE, 2022, pp. 39–50.

[2] M. Robol, T. D. Breaux, E. Paja, and P. Giorgini, "Consent verification monitoring," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 1, 2023.

[3] Merriam-Webster, "Statutory law." [Online]. Available: https://www.merriam-webster.com/legal/statutory%20law

[4] E. Johansson, K. Sutinen, J. Lassila, V. Lang, M. Martikainen, and O. M. Lehner, "Regtech – a necessary tool to keep up with compliance and regulatory changes," *ACRN Journal of Finance and Risk Perspectives, Special Issue Digital Accounting*, vol. 8, pp. 71–85, 2019.

[5] S. Abualhaija, M. Ceci, and L. Briand, "Legal requirements analysis," *arXiv preprint arXiv:2311.13871*, 2023.

[6] D. W. Arner, J. Barberis, and R. P. Buckley, "The evolution of fintech: A new post-crisis paradigm," *Geo. J. Int'l L.*, vol. 47, p. 1271, 2015.

[7] Committee of European Securities Regulators, "CESR's Guidelines on Risk Measurement and the Calculation of Global Exposure and Counterparty Risk for UCITS," 07 2010, https://www.esma.europa.eu/sites/default/files/library/2015/11/10_788.pdf.

[8] European Securities and Markets Authority (ESMA), "ESMA's Technical Advice to the Commission on the effects of product intervention measures," 02 2020, https://www.esma.europa.eu/sites/default/files/library/esma35-43-2134_technical_advice_to_the_ec_on_product_intervention.pdf.

[9] USA Government - Executive Office of the President, "88 FR 54867 - Addressing United States Investments in Certain National Security Technologies and Products in Countries of Concern," 08 2023, https://www.federalregister.gov/d/2023-17449.

[10] The European Parliament and Council, "Directive 2011/61/EU of the European Parliament and of the Council of 8 June 2011 on Alternative Investment Fund Managers and amending Directives 2003/41/EC and 2009/65/EC and Regulations (EC) No 1060/2009 and (EU) No 1095/2010," 2011. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32011L0061

[11] S. Abualhaija, M. Ceci, N. Sannier, D. Bianculli, D. Zetzsche, and M. Bodellini, "Toward automated change impact analysis of financial regulations," in *2024 IEEE/ACM 1st 1st Workshop on Software Engineering Challenges in Financial Firms (FinanSE)*. IEEE, 2024.

[12] A. Sleimi, N. Sannier, M. Sabetzadeh, L. C. Briand, M. Ceci, and J. Dann, "An automated framework for the extraction of semantic legal metadata from legal texts," *Empir. Softw. Eng.*, vol. 26, no. 3, p. 43, 2021.

[13] T. D. Breaux and A. I. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Trans. Software Eng.*, vol. 34, no. 1, pp. 5–20, 2008.

[14] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "Gaiust: supporting the extraction of rights and obligations for regulatory compliance," *Requir. Eng.*, vol. 20, no. 1, pp. 1–22, 2015.

[15] S. Ben Nasr, N. Sannier, M. Acher, and B. Baudry, "Moving toward product line engineering in a nuclear industry consortium," in *18th International Software Product Line Conference, SPLC '14*. ACM, 2014, pp. 294–303.

[16] D. G. Gordon and T. D. Breaux, "Reconciling multi-jurisdictional legal requirements: A case study in requirements water marking," in *2012 20th IEEE International Requirements Engineering Conference (RE)*. IEEE Computer Society, 2012, pp. 91–100.

[17] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, "Model-based simulation of legal policies: framework, tool support, and validation," *Softw. Syst. Model.*, vol. 17, no. 3, pp. 851–883, 2018.

[18] A. Rabinia, S. Ghanavati, L. Humphreys, and T. Hahmann, "A methodology for implementing the formal legal-grl framework: A research preview," in *Requirements Engineering: Foundation for Software Quality - 26th International Working Conference, REFSQ 2020, Pisa, Italy, March 24-27, 2020, Proceedings*, ser. Lecture Notes in Computer Science, vol. 12045. Springer, 2020, pp. 124–131.

[19] D. Torre, M. Alférez, G. Soltana, M. Sabetzadeh, and L. C. Briand, "Modeling data protection and privacy: application and experience with GDPR," *Softw. Syst. Model.*, vol. 20, no. 6, pp. 2071–2087, 2021.

[20] O. Amaral, S. Abualhaija, M. Sabetzadeh, and L. C. Briand, "A model-based conceptualization of requirements for compliance checking of data processing against GDPR," in *29th IEEE International Requirements Engineering Conference Workshops, RE 2021 Workshops*, 2021, pp. 16–20.

[21] T. D. Breaux and D. G. Gordon, "Regulatory requirements traceability and analysis using semi-formal specifications," in *Proceedings of Requirements Engineering: Foundation for Software Quality - 19th International Working Conference, REFSQ 2013*. Springer, 2013, pp. 141–157.

[22] A. Parvizimosaed, S. Sharifi, D. Amyot, L. Logrippo, M. Roveri, A. Rasti, A. Roudak, and J. Mylopoulos, "Specification and analysis of legal contracts with symboleo," *Softw. Syst. Model.*, vol. 21, no. 6, pp. 2395–2427, 2022.

[23] D. G. Gordon and T. D. Breaux, "Assessing regulatory change through legal requirements coverage modeling," in *21st IEEE International Requirements Engineering Conference, RE 2013*. IEEE Computer Society, 2013, pp. 145–154.

[24] R. Joseph, T. Liu, A. B. Ng, S. See, and S. Rai, "Newsmet: A 'do it all' dataset of contemporary metaphors in news headlines," in *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2023, pp. 10 090–10 104.

[25] V. Iyer, P. Chen, and A. Birch, "Towards effective disambiguation for machine translation with large language models," in *Proceedings of the Eighth Conference on Machine Translation, WMT 2023*. Association for Computational Linguistics, 2023, pp. 482–495.

[26] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[27] S. Abualhaija, M. Ceci, N. Sannier, D. Bianculli, L. Briand, D. Zetzsche, and M. Bodellini, *"Online Annex (online)"*, 2024, DOI: 10.5281/zenodo.10959496, available at https://doi.org/10.5281/zenodo.10959496.

[28] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Prentice Hall, 2020.

[29] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," *Advances in neural information processing systems*, vol. 32, 2019.

[30] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 255–269.

[31] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[32] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[34] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[35] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[36] F. Yu, L. Quartey, and F. Schilder, "Exploring the effectiveness of prompt engineering for legal reasoning tasks," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 13 582–13 596.

[37] M. Grüninger and M. S. Fox, "The role of competency questions in enterprise engineering," in *Benchmarking—Theory and practice*. Springer, 1995, pp. 22–31.

[38] The European Commission, "Commission Delegated Regulation (EU) No 231/2013 of 19 December 2012 supplementing Directive 2011/61/EU of the European Parliament and of the Council with regard to exemptions, general operating conditions, depositaries, leverage, transparency and supervision," 2013. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32011L0061

[39] G. H. Von Wright, *On the Logic of Norms and Actions*. Springer Netherlands, 1981, pp. 3–35.

[40] G. Boella and L. van der Torre, "Regulative and constitutive norms in normative multiagent systems," in *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning*, ser. KR'04. AAAI Press, 2004, p. 255–265.

[41] M. Ceci, T. Butler, L. O'Brien, and F. A. Khalil, "Legal patterns for different constitutive rules," in *AI Approaches to the Complexity of Legal Systems - AICOL International Workshops 2015-2017: AICOL-VI@JURIX 2015, AICOL-VII@EKAW 2016, AICOL-VIII@JURIX 2016, AICOL-IX@ICAIL 2017, and AICOL-X@JURIX 2017, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 10791. Springer, 2017, pp. 105–123.

[42] R. Hoekstra, J. Breuker, M. D. Bello, and A. Boer, "The LKIF core ontology of basic legal concepts," in *Proceedings of the 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques June 4th, 2007, Stanford University, Stanford, CA, USA*, ser. CEUR Workshop Proceedings, vol. 321. CEUR-WS.org, 2007, pp. 43–63.

[43] F. Al Khalil, M. Ceci, K. Yapa, and L. O'Brien, "SBVR to OWL 2 mapping in the domain of legal rules," in *Rule Technologies. Research, Tools, and Applications: 10th International Symposium, RuleML 2016, Stony Brook, NY, USA, July 6-9, 2016. Proceedings 10*. Springer, 2016, pp. 258–266.

[44] A. Ciabattoni, X. Parent, and G. Sartor, "Permission in a kelsenian perspective," in *Legal Knowledge and Information Systems - JURIX 2022: The Thirty-sixth Annual Conference, Maastricht, the Netherlands, 18-20 December 2023*. IOS Press, 12 2023.

[45] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 2016.

[46] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. ACL, 2002, pp. 62–69.

[47] "The difflib module," 2023. [Online]. Available: https://docs.python.org/3/library/difflib.html

[48] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, 1960.

[49] D. M. Berry, "Requirements engineering for artificial intelligence: What is a requirements specification for an artificial intelligence?" in *Requirements Engineering: Foundation for Software Quality - 28th International Working Conference, REFSQ 2022 Proceedings*, ser. Lecture Notes in Computer Science, vol. 13216. Springer, 2022, pp. 19–25.

[50] S. Harker, K. D. Eason, and J. E. Dobson, "The change and evolution of requirements as a challenge to the practice of software engineering," in *Proceedings of IEEE International Symposium on Requirements Engineering, RE 1993*. IEEE Computer Society, 1993, pp. 266–272.

[51] D. Zowghi and R. Offen, "A logical framework for modeling and reasoning about the evolution of requirements," in *3rd IEEE International Symposium on Requirements Engineering (RE'97)*. IEEE Computer Society, 1997, p. 247.

[52] C. Arora, M. Sabetzadeh, A. Goknil, L. C. Briand, and F. Zimmer, "Change impact analysis for natural language requirements: An nlp approach," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE, 2015, pp. 6–15.

[53] S. Nejati, M. Sabetzadeh, C. Arora, L. C. Briand, and F. Mandoux, "Automated change impact analysis between sysml models of requirements and design," in *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 2016*. ACM, 2016, pp. 242–253.

[54] N. Zeni, E. A. Seid, P. Engiel, and J. Mylopoulos, "NómosT: Building large models of law with a tool-supported process," *Data Knowl. Eng.*, vol. 117, pp. 407–418, 2018.

[55] T. D. Breaux and T. B. Norton, "Legal accountability as software quality: A U.S. data processing perspective," in *30th IEEE International Requirements Engineering Conference, RE 2022*. IEEE, 2022, pp. 101–113.

[56] J. C. Maxwell, A. I. Antón, and P. P. Swire, "Managing changing compliance requirements by predicting regulatory evolution," in *2012 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, September 24-28, 2012*. IEEE Computer Society, 2012, pp. 101–110.

[57] S. Saito, Y. Iimura, K. Takahashi, A. K. Massey, and A. I. Antón, "Tracking requirements evolution by using issue tickets: a case study of a document management and approval system," in *36th International Conference on Software Engineering, ICSE '14, Companion Proceedings*. ACM, 2014, pp. 245–254.

[58] S. Saito, Y. Iimura, H. Tashiro, A. K. Massey, and A. I. Antón, "Visualizing the effects of requirements evolution," in *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, Companion Volume*. ACM, 2016, pp. 152–161.

[59] D. G. Gordon and T. D. Breaux, "Managing multi-jurisdictional requirements in the cloud: towards a computational legal landscape," in *Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW 2011, Chicago, IL, USA, October 21, 2011*. ACM, 2011, pp. 83–94.

[60] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020, pp. 2898–2904.

[61] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does NLP benefit legal system: A summary of legal artificial intelligence," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 5218–5230.

[62] N. Holzenberger and B. Van Durme, "Connecting symbolic statutory reasoning with legal information extraction," in *Proceedings of the Natural Legal Language Processing Workshop 2023*. Association for Computational Linguistics, 2023, pp. 113–131.

[63] J. Soh, H. K. Lim, and I. E. Chai, "Legal area classification: A comparative study of text classifiers on Singapore Supreme Court judgments," in *Proceedings of the Natural Legal Language Processing Workshop 2019*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 67–77.

[64] E. Leitner, G. Rehm, and J. Moreno-Schneider, "Fine-grained named entity recognition in legal documents," in *International Conference on Semantic Systems*. Springer, 2019, pp. 272–287.

[65] M. Mistica, G. Z. Zhang, H. Chia, K. M. Shrestha, R. K. Gupta, S. Khandelwal, J. Paterson, T. Baldwin, and D. Beck, "Information extraction from legal documents: A study in the context of common law court judgements," in *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association*. Virtual Workshop: Australasian Language Technology Association, 2020, pp. 98–103. [Online]. Available: https://aclanthology.org/2020.alta-1.12

[66] R. Shui, Y. Cao, X. Wang, and T.-S. Chua, "A comprehensive evaluation of large language models on legal judgment prediction," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 7337–7348.

[67] A. Kwak, C. Jeong, G. Forte, D. Bambauer, C. Morrison, and M. Surdeanu, "Information extraction from legal wills: How well does GPT-4 do?" in *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, Dec. 2023, pp. 4336–4353.

[68] J. Daxenberger and I. Gurevych, "A corpus-based study of edit categories in featured and non-featured wikipedia articles," in *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*. Indian Institute of Technology Bombay, 2012, pp. 711–726.

[69] ——, "Automatically classifying edit categories in wikipedia revisions," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2013, pp. 578–589.

[70] D. Yang, A. Halfaker, R. Kraut, and E. Hovy, "Identifying semantic edit intentions from revisions in wikipedia," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2000–2010.

[71] A. Spangher, X. Ren, J. May, and N. Peng, "Newsedits: A news article revision dataset and a novel document-level reasoning challenge," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*. Association for Computational Linguistics, 2022, pp. 127–157.

[72] W. Du, V. Raheja, D. Kumar, Z. M. Kim, M. Lopez, and D. Kang, "Understanding iterative revision from human-written text," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 3573–3590.