Tracing Content Requirements in Financial Documents using Multi-granularity Text Analysis

Xiaochen Li[1*†], Domenico Bianculli[2] and Lionel Briand[3,4†]

[1*]Dalian University of Technology, Dalian, 116621, Liaoning, China.
[2]University of Luxembourg, Kirchberg, L-1855, Luxembourg.
[3]Research Ireland Lero Centre for Software Research and University of Limerick, Limerick, V94 T9PX, Ireland.
[4]University of Ottawa, Ottawa, K1H 8M5, Canada.

*Corresponding author(s). E-mail(s): xiaochen.li@dlut.edu.cn;
Contributing authors: domenico.bianculli@uni.lu; lbriand@uottawa.ca;
[†]Part of this work was done while the author was affiliated with the University of Luxembourg, Luxembourg.

Abstract

The completeness (in terms of content) of financial documents is a fundamental requirement for investment funds. To ensure completeness, financial regulators have to spend significant time carefully checking every financial document based on relevant content requirements, which prescribe the information types to be included in financial documents (e.g., the fund name, the description of shares' issue conditions and procedures). Although several techniques have been proposed to automatically detect certain types of information in documents across application domains, they provide limited support to help regulators automatically identify the text chunks related to financial information types, due to the complexity of financial documents and the diversity of the sentences typically characterizing an information type.

In this paper, we propose FITI to trace content requirements in financial documents with multi-granularity text analysis. Given a new financial document, FITI first selects a set of candidate sentences for efficient information type identification. Then, to rank candidate sentences, FITI uses a combination of rule-based and data-centric approaches, by leveraging information retrieval (IR) and machine learning (ML) techniques that analyze the words, sentences, and contexts related to an information type. Finally, using a list of domain-specific indicator phrases related to each information type, a heuristic-based selector, which considers both the sentence ranking and domain-specific phrases, determines a list of sentences corresponding to each information type.

We evaluated FITI by assessing its effectiveness in tracing financial content requirements in 100 real-world financial documents. Experimental results show that FITI is able to provide accurate identification with average precision, recall, and F_1-score values of 0.824, 0.646, and 0.716, respectively. The overall accuracy of FITI significantly outperforms the best baseline (based on a transformer language model) by 0.266 in terms of F_1-score. Furthermore, FITI can help regulators detect about 80% of missing information types in financial documents.

Keywords: Content requirements, Information type identification, Financial document, Machine Learning

1 Introduction

In the financial market, each type of investment fund, such as UCITS[1], is presented to clients through one or more financial documents (such as KIIDs — Key Investor Information Document — and prospectuses). Before these documents are made publicly available, they are submitted to national financial regulators, who check their compliance with the *content requirements* prescribed by relevant national and international laws.

The concept of "content requirement (found in the law)" has been recently proposed by Ceci et al [15], who define content requirements found in the law as "*deontic rules* requiring that some information is contained within an official document". In this work, we adopt the same definition, and consider the content requirements of financial documents, also called *financial content requirements*. For example, the following article of a national law regulating financial market [20] "*the frequency of the calculation of issue prices (should be presented)*" prescribes the inclusion of the information related to "issue price" in the prospectuses of UCITS funds.

After the submission of a financial document, agents of a financial regulator peruse the document and manually identify the passages of text (e.g., sentences or paragraphs) related to each mandated information type to ensure the document completeness, since missing information may lead to substantial fines and cause legal problems and severe investment losses when conducting activities on financial markets. However, the manual identification of information types is a non-trivial task. A financial document is usually lengthy with typically hundreds of pages and more than 3000 sentences, and contains several tables and lists. Since agents have to carefully read and analyze every sentence to avoid any misunderstanding of the content, considerable time could be spent by simply going through the entire document. The amount of manual work involved often leads to higher fund setup costs and longer time-to-market for investment funds. Therefore, it is important to develop approaches for automatically identifying the passages of text related to content requirements, which can then further enable automated compliance checking techniques.

Mining financial data play a critical role in improving the quality of financial services. Existing studies focus on mining financial data from both Web media (e.g., financial news and discussion boards [44]) and traditional financial documents (e.g., annual financial reports and 10-K [43]). In contrast to these studies, we focus on the task of tracing content requirements in financial documents, which is important to ensure their completeness. Although tracing requirements in documents have been widely studied in the areas of information extraction and software engineering [18, 29], existing approaches may not suit our task due to the complexity and the domain-specific vocabulary of financial documents. In the field of information extraction for general-purpose documents, the work typically focuses on extracting entities and relations (e.g., named entities such as persons and locations) from natural language (NL) documents [37, 41] instead of identifying sentences related to content requirements. Although recent advances in deep learning make the accurate identification of sentences possible [23, 38], the large training set required to train the underlying models [70] is usually unavailable in the financial area, due to the cost of annotating thousands of financial documents by domain experts and the differences among documents determined by national regulations. In software engineering, several studies [17, 61] infer trace links between high-level NL requirements (e.g., regulatory code) and low-level NL requirements (e.g., privacy policies). However, a typical NL requirement is often explained with one or two sentences in the regulatory text [34]; in contrast, the meaning of the same sentence in financial documents can differ across different contexts, which is seldom the case for SE requirements (where ambiguities are typically avoided). Hence, we need to design algorithms to trace financial content requirements while fully accounting for the characteristics of financial documents.

In this paper, we present FITI (Financial Information Type Identification), an approach to trace financial content requirements with multi-granularity text analysis. Its basic idea is to learn the characteristics of sentences related to an information type combining both IR and ML techniques from a small set of labeled documents. Given a new financial document, FITI selects sentences for an information type based on the analysis results of IR and ML models. Specifically, FITI first preprocesses financial documents with typical natural language processing (NLP) techniques. To conduct efficient analysis on thousands of sentences in a new financial document, a set of candidate sentences is retrieved by comparing the similarity between the

[1]UCITS (Undertakings for the Collective Investment in Transferable Securities funds) refers to a regulatory framework that allows for the sale of cross-Europe mutual funds.

related sentences in the labeled documents and every sentence in the new document. For the candidate sentences, FITI conducts a fine-grained analysis with IR and ML techniques. To capture the meaning of different sentences FITI uses similarity-based analysis with IR techniques to compare the words, sentences, and contexts between candidate sentences and the sentences related to an information type in the labeled documents. In addition, we also mine and learn a set of features relevant to an information type with feature-based analysis and train ML-based statistical models. According to the similarity- and feature-based analysis, FITI ranks each new sentence. At last, FITI uses a heuristic-based selector to select the final sentences. We built a list of domain-specific phrases that are commonly used to explain an information type (e.g., financial jargon), as well as some excluded synonyms which are seldom used to express that information type according to domain experts' suggestions and the labeled documents. By considering both sentence ranking and phrase lists, FITI identifies sentences for an information type from the candidate sentences.

We evaluated FITI using the content requirements for UCITS prospectuses. Three domain experts manually annotated sentences related to five representative information types for 100 UCITS prospectuses to form a dataset. Experimental results show FITI can accurately identify the sentences for the five information types with average precision and recall values of 0.824 and 0.646, respectively; it significantly outperforms the baselines based on keywords and language models. Further, FITI can help regulator's agents detect about 80% of missing information types. Last, FITI is effective even when the number of labeled documents is limited. With more than 40 labeled documents, the precision value of FITI is still higher than 70% for identifying most information types.

To summarize, the main contributions of this paper are:

- the first work, to the best of our knowledge, on tracing content requirements in financial documents, which is important for financial enterprises and regulators to further enable automated compliance techniques;
- the FITI approach, which addresses the problem of automated information type identification: it combines IR and ML to conduct fine-grained analysis on the sentences related to each information type;
- an extensive evaluation on the effectiveness of FITI.

The rest of the paper is organized as follows. Section 2 explains the characteristics of financial documents and their content requirements. Section 3 describes the core algorithms of FITI. Section 4 reports on the evaluation of FITI. Section 5 discusses practical implications. Section 6 surveys related work. Section 7 concludes the paper and provides directions for future work.

2 Background

2.1 Financial Documents

In the financial domain, every investment fund is required to provide informative financial documents. These documents help the financial regulator and fund clients understand all relevant and critical information of an investment. For example, KIIDs describe the nature and key risks of the fund, while a prospectus provides details about an investment offering to the public. Such financial documents mainly use natural language together with auxiliary tables and mathematical formulae. These documents are the key instruments to guarantee the compliance and controllability of an investment.

Fig. 1 shows a snippet of financial document from a prospectus. This snippet explains two types of required information in a prospectus, including *calculation method for issue price* and *issue conditions and procedures*. For example, the following sentences "The swing factor may normally not exceed 3% of the net asset value of a sub-fund . . . In such case, affected shareholders shall be informed as soon as reasonably practicable thereafter . . . " refer to the *calculation method for issue price*.

Financial documents for investment funds have several key characteristics. First, they are lengthy with typically hundreds of pages. To minimize investment risks, financial documents must provide certain elements of information required by the regulator body for any investment fund (and its sub-funds). Based on 100 randomly selected UCITS prospectuses, our statistics show that a prospectus has on average 119 pages with around 3000 sentences. Our statistics further show that an information type is usually explained with 3 to 36 sentences, on average, depending on the way an information type is explained by the investment company. These characteristics make the document difficult to read and thoroughly analyze.

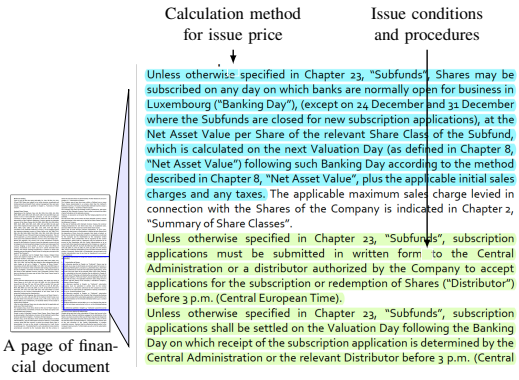


Fig. 1 A Snippet of Financial Document

2.2 Content Requirements

Before presenting them to the public, financial documents should be inspected by the regulator. One of the main inspection tasks is to check the completeness of the documents. In practice, the regulator defines content requirements for each type of financial document, specifying types of information that should be present within a document. For example, for a UCITS prospectus submitted to a national regulator, regulatory requirements stipulate about 124 information types, including for example the description of the calculation method for issue price and issue conditions and procedures (as presented in Fig. 1). The regulator must analyze the prospectus to identify the sentences discussing each information type specified by content requirements. Since missing information may cause legal problems and severe investment losses, ensuring content completeness is usually the first and most fundamental procedure before looking into the details of the financial documents. However, due to the length and peculiarities of the text in financial documents, this is usually a challenging task.

3 Information Type Identification

In this section, we present our algorithm (FITI: Financial Information Type Identification) to automatically identify the sentences related to an information type in financial documents. Its pseudocode is shown in Algorithm 1. FITI takes as input an unlabeled document d for analysis, the information type t to identify (from the content requirements), a set of labeled documents D^L (in which sentences related to t have been annotated by domain experts) with $d \notin D^L$, and some auxiliary parameters; it returns a set of sentences S from d related to t.

Algorithm 1: FITI

Input: Financial document d for analysis
Information type t
A set of labeled documents D^L, with $d \notin D^L$
Number of candidate sentences for analysis n_c
Related phrase list *rlist* and unrelated phrase list *ulist*
Threshold θ for highly similar sentences
Output: Sentences S in d related to t

```
// Step 1: Pre-processing
1  D^L, d ← preProcessing(D^L, d);
// Step 2: Identify candidate sentences
2  S^L_{r_t} ← getRelatedSentences(D^L, t);
3  S^L_{u_t} ← getUnrelatedSentences(D^L, t);
4  Candidates S^d_{c_t} ← candidateSelection(d, S^L_{r_t}, n_c);
// Step 3: Fine-grained sentence analysis
5  foreach s ∈ S^d_{c_t} do
6      s.group, s.avg_s, s.max_s, s.word
         ← calcIRSimilarity(s, S^L_{r_t}, S^L_{u_t});
7  end
8  Instances I_pos, I_neg ← getInstances(S^L_{r_t}, S^L_{u_t});
9  Classifier cls ← trainRandomForest(I_pos, I_neg);
10 foreach s ∈ S^d_{c_t} do
11     s.prob ← calcMLProbability(cls, s);
12 end
// Step 4: Sentence selection
13 n_r ← calcAvgNumOfRelatedSentences(S^L_{r_t});
14 S ← Selector(S^d_{c_t}, n_r, rlist, ulist, θ);
15 return S;
```

FITI has four main steps: pre-processing (§ 3.1), candidate sentence identification (§ 3.2), fine-grained sentence analysis (§ 3.3), and sentence selection (§ 3.4).

It first pre-processes (line 1) the document with a standard NLP pipeline (including sentence splitting, tokenization, stop words removal, stemming, named entity recognition). Then, for an information type t, FITI identifies a set of candidate sentences in d for fine-grained analysis (lines 2–4). Next, FITI analyzes the candidate sentences with information retrieval (IR) and machine learning (ML); it assigns scores to each candidate sentence (lines 5–12). Last, a heuristic-based selector is applied to select the final sentences that are most likely related to t (line 14).

3.1 Pre-processing

Financial documents are typically available in PDF format. To ease their manipulation, we convert them to a plain-text format using an off-the-shelf converter PDFBox [3]. We then apply a standard NLP pipeline to preprocess the text. The text is first split into sentences with Stanford CoreNLP [60]. Then, tokenization is applied to identify the words in a sentence. We remove the stopwords [24] and convert each word into its root form with the Porter stemming algorithm [49]. In addition, the content of financial documents often includes named entities such as numbers, person names, dates, and web addresses. To leverage the knowledge of these named entities, we perform named entity recognition [60] on the input document to generalize these named entities with their category names. Once the above steps are completed, we obtain a list of preprocessed, simplified words and sentences from the financial documents.

Application to the running example

As shown in Fig. 2, given an unlabeled document d, FITI transforms it into a list of preprocessed sentences $s_1, s_2, s_3, \cdots, s_i$. For example, a sentence "Annex I takes effect from 1 January 2016" becomes "Annex NUMBER take effect DATE". After preprocessing, the number "I" and date "1 January 2016" are transformed into "NUMBER" and "DATE", respectively.

3.2 Candidate Sentence Identification

Although financial documents include thousands of sentences, a specific information type is usually addressed by less than 50 sentences. Conducting fine-grained analysis on the entire document may therefore be impractical on commodity hardware. To solve this problem, FITI tries to efficiently filter the majority of unrelated sentences in the pre-processed document d and identify a small number of candidate sentences for further analysis. The basic hypothesis for candidate sentence identification is that sentences in d that are similar to existing sentences related to t in the labeled documents D^L may also be related to t. Therefore, we calculate similarity between sentences in d and sentences annotated as related to t in D^L. We take the top-n_c most similar sentences as candidates for fine-grained analysis.

Specifically, we collect the sentences annotated as related to t in D^L. We transform this group of sentences into a single vector (denoted as "group vector") using a standard IR model: the bag-of-words model [46]. Given a corpus (e.g., documents in D^L), the bag-of-words model gets its vocabulary (i.e., all non-duplicated words) and represents a piece of text into a vector, where the length of the vector is equal to the size of the vocabulary. In our context, each dimension of the group vector means a word in the vocabulary. If the group of related sentences does not contain a word, the value of the corresponding dimension is 0; otherwise, the value is computed by the TF-IDF (Term Frequency-Inverse Document Frequency) [46] of the word. TF-IDF is defined as:

$$\text{TF-IDF}_{w,text} = f_{w,text} \times \log \frac{N}{n_w}, \tag{1}$$

where $f_{w,text}$ denotes the number of times that w occurs in *text* (e.g., the group of related sentences), N is the number of sentences in the corpus, and n_w is the number of sentences in the corpus that contain w. TF-IDF based vectors assume that sentences can be represented with the frequently used and informative words, where TF ($f_{w,text}$) calculates the frequency of words and IDF ($\log \frac{N}{n_w}$) identifies informative words that are not used in almost all the sentences. For example, TF-IDF can identify words such as "swing" (as in "swing factor") and "dilution" (as in " dilution adjustment") in the example sentences in Section 2.1 as informative words, since they are commonly expected in financial documents but only used in specific contexts.

In this step, we do not use more complex vectorization models (e.g., deep learning based sentence embedding), as the bag-of-words model is easy to deploy and understand, it does not require a large domain-specific corpus (UCITS prospectuses in our case) to learn the embedding of domain-specific words and phrases, which is usually not available.

Further, we transform each sentence in the document d into a TF-IDF based vector as follows. For each sentence, we collect its surrounding n_{cxt} sentences, which represent the context of the current sentence. For example, if $n_{cxt} = 1$, the context of a sentence includes its previous sentence and the next sentence. We compute the frequency (i.e., TF) and the inverted document frequency (i.e., IDF) of each term in the sentence itself and its context. We transform all these TF-IDF based values into a single vector (hereafter called "context vector"). We consider the context of a sentence because a single sentence may not contain enough information

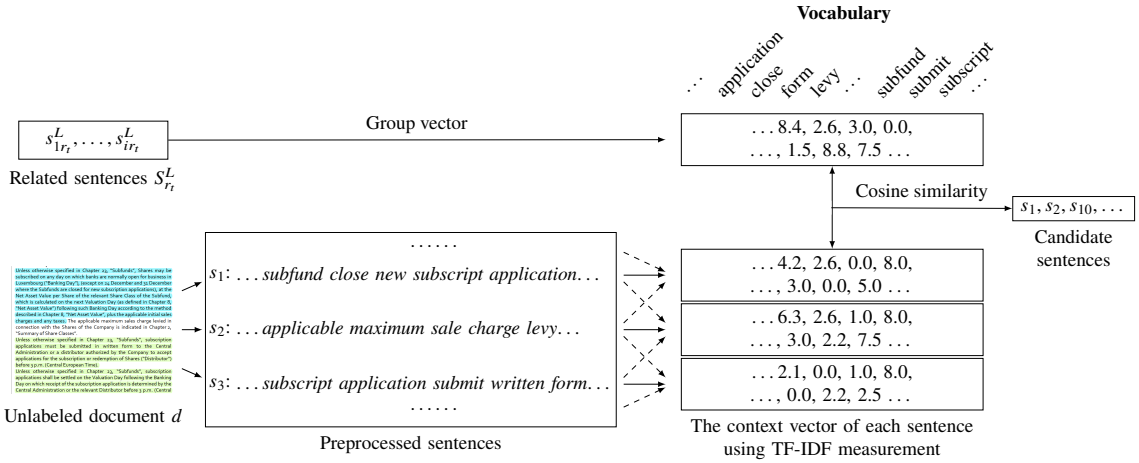


Fig. 2 Workflow of pre-processing and candidate sentence identification

for our analysis. The context of a sentence provides valuable information for understanding it.

Last, we compute the similarity between the group vector and the context vector of each sentence in d using cosine similarity [46], defined as:

$$sim(\vec{v_1}, \vec{v_2}) = \frac{\vec{v_1}\vec{v_2}}{|\vec{v_1}|\,|\vec{v_2}|}, \tag{2}$$

where $\vec{v_1}\vec{v_2}$ is the inner product of the two vectors and $|\vec{v_1}|\,|\vec{v_2}|$ is the product of the 2-norm for these vectors. We then rank and select n_c sentences in d as candidates for further analysis.

In this study, the number of surrounding sentences n_{cxt} is set to 1 and the number of candidate sentences n_c is set to 200. Our preliminary experiment shows that on average the candidate sentences contain 90% of sentences related to an information type, therefore effectively filtering 94.2% unrelated sentences and preserve the vast majority of related sentences.

Application to the running example

The workflow of candidate sentence identification is presented in Fig. 2. On the top of the figure, FITI transforms all sentences in $S^L_{r_t}$, which are annotated as related to t in D^L, into a group vector. Each dimension of the vector represents a word in vocabulary, such as "application", "close", and "form" in the example. Meanwhile, as shown at the bottom of the figure, FITI transforms each preprocessed sentence in d into a context vector. For example, when constructing the context vector for s_2, FITI computes the TF-IDF values of words in s_2, as well as in its context s_1 and s_3. All these TF-IDF values are transformed into a single vector $[\ldots 6.3, 2.6, 1.0, 8.0, \ldots, 3.0, 2.2, 7.5 \ldots]$. At last, FITI computes the cosine similarity between the vector for $S^L_{r_t}$ and each vector of the sentences in d to select candidate sentences.

3.3 Fine-grained Sentence Analysis

FITI analyzes the relevance of candidate sentences in d for an information type t (denoted as $S^d_{c_t}$) with different techniques, including both similarity-based analysis and feature-based analysis. Similarity-based analysis uses information retrieval (IR) to calculate the text similarity between a sentence in $S^d_{c_t}$ and the sentences annotated as related to t in D^L (denoted as $S^L_{r_t}$). It assumes that if a sentence is similar to the existing related sentences in $S^L_{r_t}$, this sentence is more likely to be related to t. Similarity-based analysis outputs similarity values for a sentence in $S^d_{c_t}$, which indicate the degree of text similarity of the candidate sentence with the sentences in $S^L_{r_t}$. Feature-based analysis uses machine learning (ML) to mine a set of measurable properties of sentences (features) that can distinguish related sentences from unrelated ones. It represents each sentence with a feature vector, where each component is a feature. Feature-based analysis uses the feature vectors of related and unrelated sentences in D^L to train a statistical model. For a sentence in $S^d_{c_t}$, the trained model outputs a probability value, which indicates the probability that the sentence is related to t.

Similarity- and feature-based analysis techniques analyze sentences from different perspectives: the former calculates the text similarity and the latter trains

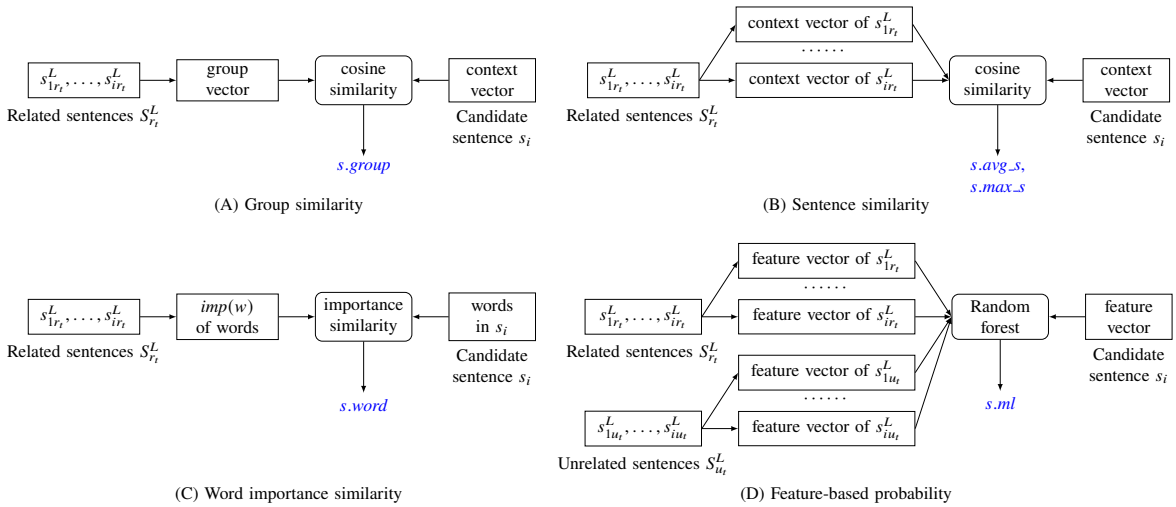


Fig. 3 Workflow of fine-grained sentence analysis

statistical models with features. They are expected to complement each other.

3.3.1 Similarity-based analysis

To perform a comprehensive comparison between sentences in $S^d_{c_t}$ and sentences in $S^L_{r_t}$, FITI calculates similarity at different granularity levels, including group similarity, sentence similarity, and word importance similarity. Since financial documents may use similar sentences to explain different information types (as explained in section 2), we use different granularity levels to better identify the candidate sentences that are similar with the overall context of $S^L_{r_t}$, individual sentences in $S^L_{r_t}$, and the words specified for t at the same time.

Group similarity compares the overall similarity between $S^L_{r_t}$ and every sentence in $S^d_{c_t}$. It is calculated as specified in Section 3.2: the group similarity is the cosine similarity between the group vector of $S^L_{r_t}$ and the context vector of a candidate sentence s (denoted as $s.group$).

Sentence similarity calculates the cosine similarity between a candidate sentence s and each sentence in $S^L_{r_t}$ based on their context vectors. Given n sentences in $S^L_{r_t}$, we can get n sentence-level similarity values for s. Based on these values, FITI calculates two sentence-level scores for s: the average and the maximum of the n similarity values, denoted as $s.avg\_s$ and $s.max\_s$, respectively. We consider $s.max\_s$ because, when a candidate sentence is extremely similar to some of the labeled sentences, it is very likely that it is also a related sentence.

Word importance similarity analyzes the importance of a word for the information type t based on the labeled documents D^L. The score of a sentence in $S^d_{c_t}$ is calculated according to word importance. This is automatically determined from three aspects; precisely, we say that the importance of a word w for t is determined by the extent to which it satisfies the three following conditions: (a) w frequently appears in the related sentence set $S^L_{r_t}$; (b) w is only present in $S^L_{r_t}$; (c) w can be found in the related sentences of every labeled document. The importance of a word w is therefore computed as:

$$\begin{aligned} imp(w) &= freq_{avg}(w) \times spec(w) \times univ(w) \\ &= \frac{1}{n} \sum_{r_i \in S^L_{r_t}} \frac{freq(r_i, w)}{|r_i|} \times \frac{\left|S^L_{r_t}(w)\right|}{\left|S^L(w)\right|} \times \frac{\left|D^L_{r_t}(w)\right|}{\left|D^L_{r_t}\right|} \end{aligned} \tag{3}$$

The first factor in the formula calculates the average frequency of w in $S^L_{r_t}$, where n is the number of sentences in $S^L_{r_t}$,$freq(r_i, w)$ counts the times that w appears in the ith related sentence r_i, and $|r_i|$ is the number of words in r_i. The second factor considers the specificity of w to t, where $\left|S^L_{r_t}(w)\right|$ is the number of sentences in $S^L_{r_t}$ that contain w, and $\left|S^L(w)\right|$ is the number of sentences in D^L containing w. Third, we analyze the universality of w in the related sentences of D^L, where $\left|D^L_{r_t}(w)\right|$ is the number of documents that have related sentences containing w, and $\left|D^L_{r_t}\right|$ is the number of documents having

related sentences. Intuitively, the importance of a word w is determined by three conditions: if w never appears in $S^L_{r_t}$, the values of these conditions are zero; in contrast, these values are 1 if w is the only word in $S^L_{r_t}$ and it never appears in other sentences. Since all the three conditions have positive correlations with the importance of a word, we multiply the values of the three conditions to reflect the importance of w. This formula is inspired by existing work on requirement analysis in software engineering [17], where they use a similar method to identify indicator terms in regulations for software requirement retrieval.

Based on $imp(w)$ for each word in D^L, the word importance similarity for a sentence s (denoted as $s.word$) is calculated as $\frac{\sum_{w \in s} imp(w)}{\sum_{w \in S^L_{r_t}} imp(w)}$. In this definition, $s.word$=0 if there is no overlapping word between s and $S^L_{r_t}$, since the $imp(w)$ values of all words in s equal zero; otherwise s is more similar with sentences in $S^L_{r_t}$ when it contains many words with high $imp(w)$ values.

3.3.2 Feature-based analysis

Feature-based analysis trains a statistical model with the sentences in D^L and uses this model to predict the probability that a sentence is related to t. It includes four main steps: training set preparation, feature engineering, model training, and prediction.

In this work, the training set is comprised of the sentences in labeled documents D^L. The positive instances for training are the related sentences $S^L_{r_t}$ for an information type t. Since the majority of sentences in D^L are unrelated, to avoid extreme imbalance, we perform under-sampling over the unrelated sentences $S^L_{u_t}$ to form the negative instances. We sample a subset of $S^L_{u_t}$ which are similar to $S^L_{r_t}$ because these sentences are expected to be more difficult to distinguish from $S^L_{r_t}$. The similarity is measured using group similarity, since it reflects the overall similarity between $S^L_{r_t}$ and a sentence in $S^L_{u_t}$. We then use ML to learn the actual distinguishing criteria. More specifically, we calculate the similarity between the group vector of $S^L_{r_t}$ and the context vector of each sentence in $S^L_{u_t}$; we select the top-ranked unrelated sentences according to the size of $S^L_{r_t}$. These sentences are textually similar with $S^L_{r_t}$. We remark that, given the writing style adopted in financial documents, sentences close to each other (e.g., in the same paragraph) usually discuss similar topics. This means that within a same paragraph that could be both related and unrelated sentences. For this reason, a sentence in $S^L_{u_t}$ is also considered as a negative instance if it is the previous or the next sentence of a related sentence.

Regarding feature engineering, we constructed 25 features for model training. Table 1 shows the name, type, and description of these features, and also their rationale. F1–10 and F11–20 are 20 features related to the important words and phrases for an information type t. F21–F25 calculate the length of a sentence and different types of named entities that could indicate the presence of an information type. With these features, each sentence in the training set can be transformed into a feature vector. The label of the feature vector is 1 or 0, representing whether a sentence is related to t or not.

In this step, we construct features based on textual information of financial documents. We do not use structural information (e.g., section headings and font formatting) for feature engineering because of two reasons. First, structural information is not always parseable. For readability reasons, in practice, a large number of financial documents are made available in Portable Document Format (PDF). A lot of structural information in such documents is missing [1], since the PDF format organizes text blocks based on the graphical coordinates of characters. Second, structural information in financial documents can be volatile [6, 56]. Companies might use their unique templates to prepare financial documents; document templates and structures can change significantly across different companies and versions. Hence, to boost the generality of FITI, we do not use structural information and only analyze textual information, which is always available in financial documents.

We trained a random forest model for each information type with its corresponding feature vectors. Random forest constructs a multitude of decision trees. Each decision tree is trained on a randomly selected subset of the training set. We use random forest because it is known to better address over-fitting on small datasets [28]; decision trees are also able to automatically identify the most discriminative features for an information type (i.e., feature selection).

During prediction, we transform candidate sentences into feature vectors and feed each of them into the trained model for the information type t, thus obtaining the probability of each sentence to be related to t (denoted as $s.prob$).

Table 1 Sentence Features for Feature-based Analysis

ID	Name (N), Type (T), Description (D), and Intuition (I)
F1–10	**(N)** Word importance **(T)** Float **(D)** We rank words by their word importance (see Section 3.3.1). We select the top-10 words as 10 features. The value of a feature is 0 if the sentence does not contain the corresponding word; otherwise, the feature value is the word importance. **(I)** These words are more important for an information type t.
F11–20	**(N)** Phrase importance **(T)** Float **(D)** We pair every two adjacent words in a sentence as a phrase. Similar to F1–10, we select the top-10 most important phrases as features. **(I)** These phrases are more important for t.
F21	**(N)** Length of a sentence **(T)** Integer **(D)** We count the number of words in a sentence. **(I)** Some information types are usually expressed with short sentences, e.g., 'Net Asset Value is calculated daily" (that explains the calculation frequency for issue price)
F22	**(N)** Ratio of 'numbers' **(T)** Float **(D)** We count how many 'numbers' in a sentence; the value is divided by the length of the sentence. **(I)** If most of the words in a sentence are numbers, the sentence is less likely to be related.
F23	**(N)** Number of 'person name' **(T)** Integer **(D)** We count the number of 'person names' in a sentence. **(I)** Some information types are associated with specific names, e.g., 'corporate name'.
F24	**(N)** Number of 'date' **(T)** Integer **(D)** We count the number of 'date' in a sentence. **(I)** Some information types are associated with dates, e.g., 'indication of date of establishment'.
F25	**(N)** Number of 'web address' **(T)** Integer **(D)** We count the number of 'web address' in a sentence. **(I)** Some information types may mention certain web addresses, e.g., 'disclaimer on periodical reports' may mention the website to retrieve the reports.

Application to the running example

Fig. 3 illustrates the way we compute similarity or probability values in the fine-grained sentence analysis. In Fig. 3(A), FITI transforms related sentences in $S^L_{r_t}$ into a group vector. This group vector is compared with the context vector of each candidate sentence s_i by cosine similarity to obtain $s.group$ for s_i. In Fig. 3(B), FITI analyzes each related sentence in $S^L_{r_t}$ independently. FITI computes the cosine similarity between the context vectors of the candidate sentence s_i and each related sentence to get $s.avg\_s$ and $s.max\_s$. In Fig. 3(C), FITI gets the importance of each word $imp(w)$ in the related sentences, which is used to compute $s.word$ of each candidate sentence based on the word importance similarity. After the similarity-based analysis, in Fig. 3(D), FITI constructs feature vectors for positive and negative instances selected from related sentences and unrelated sentences, respectively. These feature vectors are used to train a random forest model, which can output $s.ml$ based on the feature vector of each candidate sentence s_i. An example of the outputs of this step is shown in Table 2, where we compute similarity or probability values for five candidate sentences.

Table 2 Example of sentence similarity

ID	$s.group$	$s.avg$	$s.max$	$s.word$	$s.ml$	$s.score$
s_1	0.80	0.76	0.91	0.39	0.67	0.71
s_2	0.60	0.76	0.82	0.45	0.76	0.68
s_{10}	0.73	0.66	0.80	0.56	0.70	0.69
s_{12}	0.82	0.73	0.88	0.65	0.66	0.75
s_{16}	0.75	0.76	0.81	0.49	0.67	0.70

3.4 Sentence Selection

For an information type t, FITI selects sentences from a document d according to Algorithm 2. The basic idea is to select sentences that are either highly similar to $S^L_{r_t}$ regarding at least one aspect (i.e., the overall level, the individual sentence level, or the important word level) or ranked higher by the comprehensive score decided by both similarity- and feature-based analysis. If we combine the above with key-phrase lists, we can refine the selection of sentences and accurately decide the final related sentence set. The inputs include the candidate sentences $S^d_{c_t}$ and their similarity and probability scores, the average number of related sentences per labeled document n_r, a list of domain-specific related phrases *rlist* and unrelated phrases *ulist*, and the auxiliary parameter θ. Notice that *rlist* summarizes the phrases frequently used to express t; *ulist* contains the phrases that are synonyms with related phrases but are seldomly used to express t. Since domain experts usually use keyword search to help them find the possible location of related sentences, these lists can be manually constructed when deciding the criteria to annotate the training documents D^L.

Before sentence selection, FITI detects duplicate sentences in $S^d_{c_t}$ (line 1). Two sentences are considered as duplicates if the similarity between their context vectors is larger than a threshold θ. Duplicate sentences are similar and usually express the same semantic meaning. FITI will either select or exclude them together.

FITI first selects sentences by their similarity scores. A sentence (and its duplicates) is selected if at least one of its similarity score ($s.group$, $s.avg\_s$,

Algorithm 2: Selector

Input: Candidate sentences $S^d_{c_t}$ with $s.group$, $s.avg\_s$, $s.max\_s$, $s.word$, and $s.ml$ of each sentence
Average number of related sentences per document n_r
Related phrase list *rlist* and unrelated phrase list *ulist*
Threshold θ for highly similar sentences
Output: Sentences S in d related to t

1 $S^{d\,\prime}_{c_t} \leftarrow mergeSimilar(S^d_{c_t}, \theta)$;
2 $S \leftarrow selectBySimilarity(S^{d\,\prime}_{c_t}, \theta)$;
3 $S^{d\,\prime}_{c_t} \leftarrow S^{d\,\prime}_{c_t} \backslash S$;
4 **foreach** $s \in S^{d\,\prime}_{c_t}$ **do**
5 $s.score \leftarrow (s.group + s.avg\_s + s.max\_s + s.word + s.ml)/5$;
6 **end**
7 $S^{rank}_{c_t} \leftarrow rankByScore(S^{d\,\prime}_{c_t})$;
8 $i = 0$;
9 **while** $|S| < n_r$ *and* $i < |S^{rank}_{c_t}|$ **do**
10 $S = S \cup S^{rank}_{c_t}[i++]$;
11 **end**
12 $S_g \leftarrow groupByDistance(S)$;
13 $S \leftarrow \emptyset$;
14 **foreach** $s_g \in S_g$ **do**
15 Boolean $rCheck \leftarrow hasWordsInList(s_g, rlist)$;
16 Boolean $uCheck \leftarrow hasWordsInList(s_g, ulist)$;
17 **if** $rCheck$ *and* $!uCheck$ **then**
18 $S = \{s_g\} \cup S$;
19 **end**
20 **end**
21 return S;

$s.max\_s$, or $s.word$) is greater than θ, because this sentence may express the same meaning as some related sentences in D^L. As for unselected sentences, we calculate a sentence score for each sentence according to its similarity (from similarity-based analysis) and probability (from feature-based analysis) scores (lines 4–6). We rank sentences by their sentence scores and select the top-ranked sentences (and their duplicates) until the number of selected sentences reaches n_r (lines 7–11).

Last, we group the selected sentences based on their position in the document, because information types are usually addressed by several continuous sentences. We put any two sentences into a group if the distance between them is less than three sentences, since these sentences usually share the same context. For example, we put sentences s_i and s_{i+2} into a group as they have the same context sentence s_{i+1}. For each group of sentences, we check whether they contain domain-specific phrases in the related list *rlist* or unrelated list *ulist*. We annotate a group of sentences as related if they feature phrases in *rlist* but no phrase in *ulist* (lines 14–20).

Application to the running example

In this step, we assume to configure FITI to select four sentences from the five candidate sentences listed in Table 2. First, FITI detects duplicate sentences. In this example, no sentence pairs are duplicate. Then, FITI selects sentences by their similarity scores. In this example, we set the threshold θ to 0.9. Therefore, s_1 is selected. For the remaining four sentences, we rank them based on $s.score$, and select the top-three sentences (i.e., s_{12}, s_{16}, and s_{10}). Third, FITI groups the selected four sentences based on their position. We get three groups, which are s_1, s_{10}–s_{12}, and s_{16}. FITI decides the final sentences by comparing words in each group with *rlist* and *ulist*.

4 Evaluation

In this section, we evaluate our approach (FITI) for financial information type identification. First, we assess the accuracy of FITI in identifying the sentences related to an information type. Then, we evaluate the factors that impact this accuracy, including different AI techniques (i.e., similarity-based analysis with IR and feature-based analysis with ML), the size of the training set, and the sentence selection strategy. Last, we analyze how FITI helps inspect financial documents. More specifically, we answer the following research questions:

RQ1 *Can FITI accurately identify the information types in financial documents?*
RQ2 *How do different AI techniques affect the accuracy of FITI?*
RQ3 *What is the impact of the size of labeled documents on the accuracy of FITI?*
RQ4 *What is the impact of the sentence selection strategy on the accuracy of FITI?*
RQ5 *How can FITI support the compliance analysis of financial documents?*

4.1 Dataset and Settings

We evaluated FITI with the content requirements for UCITS prospectuses [20], because UCITS is one of the most popular and representative investment regulatory

Table 3 Basic Statistics on the Dataset

Item	Value
Num. of prospectuses	100
Years of publishing	2010–2021
Num. of pages (avg. / range)	119 / 36–547
Num. of sentences (avg. / range)	2968 / 683–15458
Num. of sentences annotated (avg. / range)	T1: 3.6 / 1–12 T2: 5.6 / 1–29 T3: 17.8 / 11–54 T4: 15.9 / 6–27 T5: 36.4 / 4–97
Num. of phrases in *rlist*	151
Num. of phrases in *ulist*	82

frameworks, which has over €10 trillion of assets under management across the world [27].

We randomly collected 100 approved UCITS prospectuses from the official website of a national regulator[2] as our dataset. The basic statistics on the dataset are shown in Table 3. The dataset contains prospectuses published between 2010 and 2021, covering all the calendar years since the establishment of the "UCITS Law 2010". The number of pages of these prospectuses varies significantly from 36 to 547, with an average of 119. In these prospectuses, the number of sentences ranges from 683 to 15 458. On average each prospectus has 2968 sentences.

All the documents were annotated by three domain experts. Due to the time required for annotating documents, the domain experts selected five representative information types for evaluation, including *disclaimer on periodical reports* (T1), *calculation frequency for issue price* (T2), *calculation method for issue price* (T3), *liquidation conditions and procedure* (T4), and *issue conditions and procedure* (T5). They selected these information types by considering their importance, complexity, and diversity. First, all these information types are content requirements, which require to be present — with explanatory sentences — in every prospectus. Second, manually identifying these information types is time-consuming, as one has to identify relevant sentences among, on average, around 3000 sentences. Third, these information types are diverse in terms of the average number of related sentences (as shown in Table 3, we have 3.6 for T1, 5.6 for T2, 17.8 for T3, 15.9 for T4, and 36.4 for T5). For example, the information type T1 has at most 12 related sentences per prospectus; in contrast, the information type T5 can be related to up to 97 sentences. These information types are also diverse in terms of the wording and writing styles. For example, T1 (*disclaimer on periodical reports*) is usually explained by only a few sentences with key phrases illustrating the places and the charge to obtain the reports, while T5 (*issue conditions and procedure*) can be discussed with many long sentences, demonstrating different conditions and procedures to process the issue (e.g., reject subscriptions, limit/restrict the issue). Overall, this selection allowed us to assess how FITI identifies information types specified at different levels of details (i.e., from a single sentence to several pages of sentences).

The annotation was conducted in two phases. First, the domain experts selected 50 documents from the dataset. They perused these documents to define the detailed criteria for annotation (i.e., what types of sentences/phrases should be related/unrelated to an information type). In this phase, one of the domain experts first defined the initial annotation criteria. For example, the criteria to annotate a sentence as T1:*disclaimer on periodical reports* are: (1) "the sentence indicates where the periodical reports may be obtained (e.g., a website or the registered office of the management company)"; (2) "the sentence indicates that the reports can be obtained free of charge"; (3) "the sentence describes the periodical reports (e.g., their frequency) and is in the same paragraph with one of the sentences satisfying criterion (1) or (2)". The expert defined and consolidated the initial criteria over two weeks; this time frame includes the time to define the individual annotation criteria for each information type as well as the time to resolve interdependencies among criteria. The initial criteria were then sent to the other two domain experts. They had four 2-hour workshops in two weeks to further refine the criteria. The main issues discussed during such workshops were *vague criteria* and *missing criteria*. As shown in the top row of Table 4, vague criteria (i.e., criteria for which the initial description was vague) were refined by improving the their textual description; when the experts noticed that a criterion could not cover all related sentences or could lead to the inclusion of some unrelated sentences for a given information type, they added new inclusion or exclusion criteria for that information type. For example, for T1, they added the exclusion criterion "we do not annotate sentences that describe the periodical reports but are in different paragraphs from the sentences satisfying criterion (1) or (2)" to avoid false positive annotations. Overall, the first phase of annotation took about one month.

[2]CSSF approved prospectuses. https://www.bourse.lu/home

Table 4 Issues to be solved during the two Phases of the Annotation Process

Phase	Issue Type	Meaning	Solution
Phase 1: Determining annotation criteria	Vague criterion	The description of an initial criterion is vague	Improve the text of the criterion to avoid misunderstandings
	Missing criterion	A criterion cannot cover all related sentences or may lead to the inclusion of some unrelated sentences	Add new inclusion or exclusion criteria for each information type
Phase 2: Annotating information types	Missing annotation	A part of related sentences is not annotated	Cross-check the annotations to identify the missing parts
	Unrelated annotation	Some sentences are partially related	Discuss whether to add or delete these sentences case by case; more annotation criteria can be created to ensure agreement among the annotators

During the second phase of the annotation, domain experts annotated all 100 documents with the selected information types based on the established annotation criteria. Each person annotated a disjoint subset of documents individually. In this phase, we allocated three weeks for all the domain experts to complete their initial annotation. They then examined each other's annotations. As part of this step, five 2-hour workshops were conducted over five weeks to discuss possible incorrect annotations and fix them when warranted. The two main types of issue detected in this step, shown in the bottom part of Table 4, were *missing annotation* and *unrelated annotation*. Missing annotations occurred because some information types (e.g., T5) were associated with dozens of sentences spread across different pages; domain experts could easily miss to annotate some of these related sentences. Such issues were solved by cross-checking the annotations to identify the missing parts. An unrelated annotation indicated that a domain expert had wrongly marked some sentences as related; this happened because some paragraphs were only partially related to an information type. Issues of this type were resolved by discussing, case by case, whether to add or delete sentences, and by adding or removing annotation criteria to ensure an agreement among the annotators. For example, in the case of T3:*calculation method for issue price*, although the documents include some sentences mentioning "issue price", they do not provide the details for the calculation. Domain experts finally considered such sentences as unrelated, and added a new exclusion criterion for T3 "we do not capture sentences generically indicating how the sales or subscription price, or the net asset value, are calculated, if they do not provide — or reference — the details of the calculation". Overall, the second phase of annotation took in total about two months.

We remark that we did not consider a larger number of information types due to the complexity of and the time required by the annotation task. On the one hand, domain experts have to identify the related sentences from thousands of sentences in a prospectus. On the other hand, an information type can be related to multiple sentences, which are usually distributed across different pages. This means that domain experts need to identify and double-check all these sentences.

The phrase lists *rlist* and *ulist* (see § 3.4) used for sentence selection were manually built based on the phrases listed in the annotation criteria. We extended the phrases with synonyms occurring in the first 50 documents. There are in total 151 phrases in the *rlist* and 82 phrases in the *ulist*. We set the parameter θ to 0.9; it was decided empirically by evaluating the accuracy of FITI using a range of values between 0.1 and 1 with a step of 0.1 on the first group of 50 documents.

We performed the experiments with a computer running macOS 11.1 with a 2.30 GHz Intel Core i9 processor and 32GB memory.

4.2 Accuracy of FITI (RQ1)

To answer RQ1, we assessed the accuracy of FITI in identifying sentences related to different information types.

4.2.1 Methodology

We evaluated FITI using the annotated documents with k-fold cross-validation ($k = 5$). Since 50 annotated documents were used for phrase list construction and parameter tuning, we kept them in the training set and

Table 5 Accuracy of Information Type Identification Algorithms

ID	Precision				Recall				F_1-score			
	KW	BERT	$BERT_{KW}$	FITI	KW	BERT	$BERT_{KW}$	FITI	KW	BERT	$BERT_{KW}$	FITI
T1	0.100	0.407	0.743	0.855	0.682	0.543	0.558	0.722	0.176	0.466	0.512	0.783
T2	0.553	0.301	0.797	0.788	0.463	0.658	0.353	0.536	0.504	0.413	0.490	0.638
T3	0.597	0.258	0.672	0.784	0.326	0.682	0.209	0.662	0.422	0.375	0.318	0.718
T4	0.772	0.485	0.864	0.882	0.615	0.941	0.403	0.893	0.685	0.640	0.549	0.887
T5	0.669	0.275	0.696	0.812	0.293	0.511	0.190	0.418	0.407	0.358	0.298	0.552
Avg	0.539	0.345	0.700	0.824	0.476	0.667	0.343	0.646	0.439	0.450	0.433	0.716

only test FITI on the remaining 50 documents. In each fold, we selected 10 documents from the remaining 50 documents as the test set; the training set included the other 90 documents. Given an information type t and a test set, we compared the sentences selected by FITI with the ground truth annotated by the domain experts. We measured the accuracy of FITI with *precision*, *recall*, and F_1-*score* (F_1). They are defined as $Precision = \frac{|TP|}{|TP|+|FP|}$ and $Recall = \frac{|TP|}{|TP|+|FN|}$, where true positives (TP) and false positives (FP) refer to sentences selected by FITI which are related or not to t, respectively. False negatives (FN) refer to cases where FITI misses a sentence related to t. F_1-*score* is defined as $F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$. These evaluation metrics work at the sentence level. For example, when there are four consecutive sentences annotated by the domain experts in the ground truth, and FITI correctly identifies three of them, we have $TP = 3$, $FP = 1$, and $Precision = 0.75$.

We compared FITI with three baselines: a keyword-search strategy (denoted as KW), the Bidirectional Encoder Representation from Transformers (BERT) language model, and a combination of BERT and KW (denoted as $BERT_{KW}$).

KW is a common way for regulator agents to locate sentences. Since we have constructed two phrase lists containing related phrases (i.e., *rlist*) and unrelated phrases (i.e., *ulist*), KW directly analyze the candidate sentences to select those that feature phrases in *rlist* and no phrase in *ulist* as related to an information type.

BERT is a transformer-based language model for NLP [22, 55]. We take BERT as a baseline because it is a typical deep neural network that has been widely used for requirement mining in requirement engineering [63] The input of BERT is a sentence for classification and its context sentences (i.e., its previous sentence and the next sentence, as explained in Section 3.2), which is the same information used by FITI. In the training phase, we fine-tuned a pre-trained BERT for each information type with the training set constructed in Section 3.3.2 for feature-based analysis of FITI. In the testing phase, BERT predicted the relatedness of each candidate sentence to an information type.

$BERT_{KW}$ is a combination of BERT and KW. For each sentence predicted as related by BERT, we improved the prediction based on key-phrase lists. Among the sentences predicted by BERT as related, we further retained the sentences that feature phrases in *rlist* but no phrase in *ulist* as the final set of sentences related to an information type.

We implemented KW from scratch. BERT was implemented with the open source Python library for transformers[3]. BERT was pretrained with the `bert-base-uncased` dataset. Hyper-parameters were fine-tuned using early stopping and the Adam optimizer based on the cross entropy loss [16]. In each iteration of fine-tuning, we set the batch size to 64. Fine-tuning stopped when the loss did not change for 10 iterations.

4.2.2 Results

Table 5 shows the accuracy of the different information type identification algorithms. FITI identified a set of sentences related to each information type with a precision value ranging from 0.784 to 0.882. The differences in precision between these information types are less than 10%. The precision value for T4 is the highest (0.882). We found that prospectuses tend to use similar sentences to explain the liquidation conditions and procedure (T4); therefore, all the algorithms got a relatively high precision score on this information type. The precision values of T2 and T3 are relatively low, but still near 80% for FITI. The recall value of FITI ranges from 0.418 to 0.893, with an average recall value of 0.646. The result means that FITI could find, on average, more than 60% of sentences for an information type. Among these information types, the recall

[3]Hugging Face Transformers https://github.com/huggingface/transformers

values of T2 and T5 are low. For T2, the low recall value is caused by the small number of sentences related to T2. When a related sentence is not identified by FITI, recall could change dramatically. For T5, there are on average 36.4 sentences related. FITI may not easily identify all of these sentences, leading to low recall.

KW performed poorly compared to FITI, identifying an average of 0.476 related sentences with an average precision value of 0.539. FITI outperformed KW by 0.277 (0.716 vs 0.439) in terms of F_1-score. Although keyword search is a common activity for regulator agents, many false positives can be returned due to the case that sentences related to different information types share a common vocabulary. Moreover, since the writing styles and the number of related sentences can differ across financial documents from different investment companies, regulator agents may not enumerate all keywords for every related sentence, leading to a low recall value. In contrast, FITI could leverage these (possibly incomplete) keyword lists to improve its accuracy in identifying information types.

As for the transformer language model BERT, its average recall value is 0.667, which is similar to the one obtained by FITI. BERT has a higher recall than FITI for information types T2–T5. However, the precision value of BERT is much lower than that of FITI (i.e., 0.345 vs 0.824), since BERT wrongly predicts many sentences as related. When integrating the key-phrase lists into BERT, BERT_{KW} can identify information types more accurately. The average precision of BERT_{KW} improves from 0.345 (the value achieved by BERT) to 0.700. However, as a trade-off, the average recall of BERT_{KW} drops dramatically to 0.343. As a result, FITI outperforms both BERT and BERT_{KW} in terms of F_1-score. The results obtained by BERT can be explained as follows. In this task, we need to account for the limited size of domain-specific datasets, bounded by the high cost of annotations, which must be performed by domain experts. Numerous neural network weights in a deep neural network like BERT may not be well fine-tuned in this context. In addition, as discussed in Section 2, financial documents can use similar sentences to explain different types of required information. Hence, many sentences are classified by BERT as false positives, leading to a low precision value.

We conducted the Wilcoxon test on the prediction outputs of the different algorithms in terms of the F_1-score obtained for each information type in the testing documents; we chose this test since it is non-parametric and does not require any distributional assumption [21]. The results confirm that the differences in the prediction between FITI and the baselines are statistically significant (p-value < 0.05).

The current results of FITI can be interpreted as follows. With an average precision above 80%, FITI can help users efficiently locate the correct position of 64.6% related sentences. Compared to analyzing the whole document, users can use FITI to reduce the effort in manually finding sentences related to an information type. The current results are also important for FITI to be used in the context of (financial) document compliance checking. In compliance checking the existence of an information type can be established if FITI can identify at least one sentence for this information type. With a high precision value, it means that when FITI finds some sentences, they are usually the actual related sentences for an information type. Therefore, FITI can correctly decide the existence of this information type. The high precision value is also important to detect missing information types. FITI can confidently (i.e., with high precision) find related sentences for an information type. These sentences are frequently used to explain an information type. However, given a financial document, if FITI cannot find any of such sentences, it is more likely that the financial document misses this information type.

Performance analysis

We have further analyzed the predictions made by FITI, to identify the cases in which it performs well as well as those with subpar performance. We have identified four cases:

Case 1 (Positive): *FITI can correctly identify information types when they are explained with similar sentences in different prospectuses.* We found that for some information types, prospectus writers tend to use similar structures and sentences to explain them. A typical example is T4: *liquidation conditions and procedure*, where FITI and all baselines got the highest F_1-score on the dataset. Since FITI conducts multi-granularity similarity analysis (i.e., group similarity, sentence similarity, and word importance similarity) on the documents, the similarity across different documents can be correctly identified by FITI.

Case 2 (Positive): *FITI performs well when the sentences related to information types contain certain named entities or keywords.* FITI uses feature-based analysis to capture important words, phrases, and named entities (i.e., number, person name, date, and web address). When the sentences of an information

type are associated with such keywords and named entities, FITI could distinguish these sentences from those related to other information types. For instance, information type T1:*disclaimer on periodical reports* indicates where the periodical reports may be obtained. It is usually associated with (web) addresses for obtaining the periodical reports. The feature-based analysis of FITI can correctly identify these sentences, increasing prediction accuracy.

Case 3 (Negative): *FITI may select sentences that are only partially related to an information type.* As discussed in Section 4.1, some paragraphs may be partially related to an information type, which means that they mention an information type but do not provide all the corresponding details. These partially related sentences (e.g., T3:*calculation method for issue price*) are difficult to label even for domain experts. FITI may also include these partially related sentences in the prediction results, leading to false positives. A structure-based analysis of the documents, to be conducted as part of future work, could reduce the false positives. Although a concept (e.g., issue price) can be discussed in different sections, some sections (e.g., the background) will only discuss an overview of the concept without going into the details. By analyzing the structure of prospectuses, some false positives could be filtered.

Case 4 (Negative): *It is difficult for FITI to identify the sentences related to similar information types.* Some information types are similar, such as T2:*calculation frequency for issue price* and T3:*calculation method for issue price*. They are both related to "issue price", though the focus of each of them is on different topics (i.e., calculation frequency and calculation method). Since these two information types are related and usually explained together, FITI may not correctly distinguish between them, leading to low accuracy compared to other information types (as shown in Table 5). To increase the accuracy, as part of future work, topic models can be used to analyze the different topics discussed in similar information types.

To conclude, *the answer to RQ1 is that FITI identifies an average of 64.6% of relevant sentences for an information type, with an average precision value of 0.824, significantly outperforming the baselines based on keywords and language models.*

4.3 Impact of Different Components (RQ2)

FITI selects related sentences based on both similarity- and feature-based analyses with IR and ML techniques (section 3.4). To answer RQ2, we assessed the impact of these two types of analysis on the accuracy of FITI.

4.3.1 Methodology

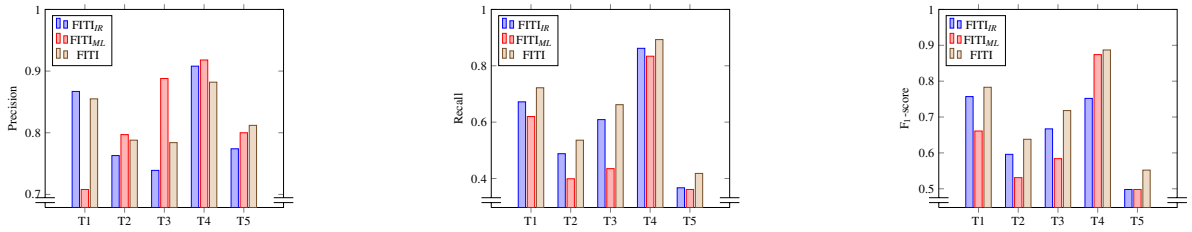
We implemented two variants of FITI (called FITI_{IR} and FITI_{ML}) to assess the possible impact of each technique. During sentence selection, FITI_{IR} only selects sentences based on the similarity values calculated in the similarity-based analysis, while FITI_{ML} performs sentence selection only relying on the probability calculated in the feature-based analysis. To implement FITI_{IR}, we calculated the score of a sentence by averaging the four similarity values (i.e., $s.group$, $s.avg\_s$, $s.max\_s$, $s.word$) at line 5 in Algorithm 2. To implement FITI_{ML}, we disabled the function *selectBySimilarity* at line 2 and assigned the score of a sentence as the probability value calculated from the feature-based analysis ($s.ml$). We ran the standard version of FITI (i.e., the one presented in section 3) and the additional variants using the same settings as in RQ1.

4.3.2 Results

As presented in Figure 4, similarity- and feature-based analysis show different abilities in analyzing information types.

Feature-based analysis (FITI_{ML}) tends to assign a high probability value to a small fraction of related sentences. Hence, FITI_{ML} achieves higher precision values, for the majority of information types (i.e., T2 to T5), than FITI_{IR}. The precision values of T2, T3, and T4 are also higher than those of FITI (in Figure 4a). However, the recall value of FITI_{ML} is lower than both FITI_{IR} and FITI (in Figure 4b).

The above results can be explained as follows. FITI_{ML} analyzes sentences based on features related to important words/phrases, length of sentences, and named entities (as shown in Table 1). Since these features capture the characteristics of information types, FITI_{ML} identifies a subset of related sentences that best reflect these features with high precision. For example, FITI_{ML} can identify sentences containing the information of calculation date and important words for information type T2:*calculation frequency for issue price*. An exception is T1, for which the precision value



(a) Precision of FITI and its variants (b) Recall of FITI and its variants (c) F_1-score of FITI and its variants

Fig. 4 Impact of similarity-based and feature-based analyses for FITI

is low. We conjecture the reason is the small number of positive instances (i.e., 3.6 related sentences per document on average) for training the ML model.

As to similarity-based analysis (FITI$_{IR}$), it tends to retrieve more related sentences than FITI$_{ML}$, leading to a high recall, though some are false positives. FITI$_{IR}$ is an instance-based algorithm that retrieves many sentences similar to ground truth sentences. However, as discussed in Section 2, financial documents may use similar sentences to explain different types of required information. Many unrelated similar sentences could also be included, leading to a high recall value but a low precision value.

By integrating the two components, the accuracy in identifying information types is further improved: FITI achieves a higher F_1-score than both FITI$_{IR}$ and FITI$_{ML}$ with p-value < 0.05 regarding the five information types (in Figure 4c). When integrating the two components, more related sentences identified by either IR or ML are included, as well as some unrelated sentences (e.g., similar sentences that explain other information types). Hence, the final recall value is improved, but the precision value is sometimes lower than that obtained by the other variants. These results show that the integration of the components is necessary. After integration, FITI achieves a substantial improvement in terms of recall compared with FITI$_{IR}$ and FITI$_{ML}$, with an average precision value that is only less than 2% lower than that of FITI$_{ML}$.

The answer to RQ2 is that both similarity-based and feature-based analyses contribute to improving the accuracy of FITI, complementing each other.

4.4 Impact of the Size of the Training Set (RQ3)

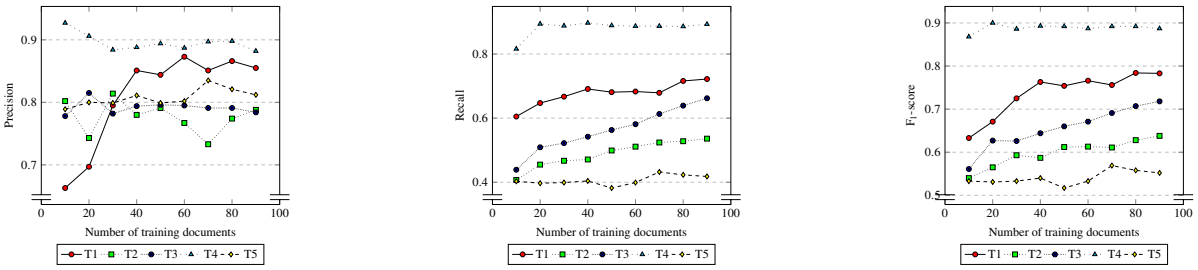
FITI conducts information type identification relying on the documents in the training set annotated by domain experts. This RQ assesses the impact of the size of the training set on the accuracy of FITI. It is an important question as access to such annotated documents is, in practice, inherently limited.

4.4.1 Methodology

We evaluated the accuracy of FITI by varying the size of the training set from 10 to 90 documents in steps of 10. For each training set size, we built the training set incrementally: each time we randomly selected 10 documents and added them to the training set. In other words, for each training set size, the training set is a superset of the one used for the previous value. For instance, the training set for size 40 was obtained by randomly selecting 10 documents and adding them to the training set for size 30. We trained FITI on each sampled training set and used the trained model to identify sentences in the test set for different information types. We measured the accuracy of FITI for different training set sizes, as when addressing RQ1. We used 5-fold cross validation, so we repeated this process five times for each training set size.

4.4.2 Results

As shown in Figure 5, precision largely increases (i.e., in the case of T1) or becomes relatively stable within a certain range (i.e., in the case of T2, T3, T4, and T5) as the size of the training set increases. We found that with 10 to 90 documents in the training set, the precision for information types T1 and T2 fluctuates more; as shown in Table 3, there are only 3.6 to 5.6 sentences on average related to these information types. When a related or an unrelated sentence is added to the prediction results, the precision can thus change a lot. For T3–T5, the fluctuation on precision is small. The changes in precision (e.g., at the 70–80 documents threshold) are mainly caused by two reasons. First, we randomly added 10 more documents to the training set each time; the selection of these new documents affects precision. Second, when FITI selects more sentences to increase recall, as a complementary evaluation metric,



(a) Precision of FITI (b) Recall of FITI (c) F_1-score of FITI

Fig. 5 Impact of the size of the training set for FITI

precision on the 10 documents in the testing set could be affected.

With more than 40 annotated documents, precision is stable, varying within a 10% range for all information types. Further, we observe that the precision value is still higher than 70% for most information types (i.e., in the case of T2, T3, T4, and T5) when there are only 10 annotated documents. This is because FITI is able to identify a small fraction of sentences that are highly similar to the annotated related sentences in the training set. However, as shown in Figure 5b, the recall value of FITI is low; for information types T2, T3, and T5, recall is around 40%. As more documents are annotated, recall largely increases, because FITI may learn more about different expressions or wordings to explain an information type. F_1-score also confirms the positive impact of increasing the size of the training set on FITI.

The answer to RQ3 is that the accuracy of FITI improves as the size of the training set increases. FITI requires a minimum size of 40 annotated documents to achieve high and stable precision.

4.5 Impact of the Sentence Selection Strategy (RQ4)

FITI selects sentences related to an information type based on a list of domain-specific related phrases and a list of unrelated phrases (*rlist* and *ulist* in Algorithm 1, respectively). We expect these two lists to be manually built by domain experts when defining the criteria for annotating the training set of financial documents. However, though a one-off task, it could cause significant burden to the domain experts, in this RQ we analyze alternative sentence selection strategies to assess their impact on the accuracy of FITI.

4.5.1 Methodology

We compared the sentence selection strategy in FITI (which ranks sentences based on their sentence scores computed by the IR and ML techniques) with three alternative strategies that are commonly used for selecting items from a ranked list:

- FITI$_{score}$ selects only sentences in a ranked list whose scores are above a user-defined threshold θ_{score};
- FITI$_{topN}$ selects the top-N sentences in a ranked list, where N is an input parameter;
- FITI$_{avgN}$ sets a different N for each information type, which is automatically determined based on the average number of sentences related to this information type in all financial documents in the training set; FITI$_{avgN}$ uses this number to select the top-ranked sentences as related.

We replaced the sentence selection strategy in FITI with the three above alternative strategies, and compared the accuracy of such FITI variant for information type identification. Since FITI$_{score}$ and FITI$_{topN}$ have additional input parameters (i.e., θ_{score} and N, respectively), we decided their values by conducting a preliminary evaluation, aimed at identifying the parameter value of FITI$_{score}$ and FITI$_{topN}$ leading to the highest F_1-*score* for the majority of information types. We varied θ_{score} from 0 to 1 with a step of 0.05: the highest F_1-*score* was obtained when θ_{score} was set to 0.4. We varied N from 1 to 5 with a step of 1, obtaining the highest F_1-*score* when N was set to 3. We used these values ($\theta_{score} = 0.4$ and $N = 3$) in our experiments.

4.5.2 Results

Table 6 shows the accuracy results (in terms of precision, recall, and F_1-score) when using the different sentence selection strategies; the value in bold is the best result achieved for a given evaluation metric.

Table 6 Accuracy of Different Sentence Selection Strategies

ID	Precision				Recall				F_1-score			
	$FITI_{score}$	$FITI_{topN}$	$FITI_{avgN}$	FITI	$FITI_{score}$	$FITI_{topN}$	$FITI_{avgN}$	FITI	$FITI_{score}$	$FITI_{topN}$	$FITI_{avgN}$	FITI
T1	0.713	0.712	0.713	**0.855**	0.740	**0.749**	0.744	0.722	0.726	0.730	0.728	**0.783**
T2	0.407	0.483	0.401	**0.788**	0.440	**0.635**	0.440	0.536	0.423	0.549	0.420	**0.638**
T3	0.706	0.744	0.636	**0.784**	0.642	0.606	**0.669**	0.662	0.673	0.668	0.652	**0.718**
T4	**0.924**	0.860	0.834	0.882	0.814	0.855	0.851	**0.893**	0.866	0.857	0.842	**0.887**
T5	0.612	0.704	0.468	**0.812**	0.351	0.334	**0.429**	0.418	0.446	0.453	0.448	**0.552**
Avg	0.672	0.701	0.610	**0.824**	0.597	0.636	0.627	**0.646**	0.627	0.651	0.618	**0.716**

From the table, we can see that the original sentence selection strategy used by FITI (the one based on phrase lists) leads to the best precision value for four out of five information types (and yields the second-best result for the fifth information types, T4), with an average precision value of 0.824, outperforming the three alternative strategies with a difference ranging from 0.123 (0.701 vs 0.824) to 0.214 (0.610 vs 0.824). The reason for such a high precision value is that the phrase lists provide extra domain knowledge on related and unrelated sentence patterns, which are important for FITI to accurately assess how related (to a certain topic) a list of sentences is.

Regarding recall, although the average recall value of FITI is higher than those obtained when using the three alternative strategies, FITI achieves the highest recall value only for one information type (T4). The reason is that, when leaving sentences out based on the unrelated phrase list *ulist*, a few related sentences could be incorrectly dismissed. The difference between the recall value of FITI and the best recall value is less than 3% for T1, T3, and T5, and goes up to 10% for T2. As discussed in Section 4.2, in the context of (financial) document compliance checking, it is desirable to use identification algorithms that yield a high precision value. Hence, a slight decrease in recall is acceptable, considering the high precision value achieved by FITI with the sentence selection strategy based on phrase lists. This is also reflected when looking at the overall accuracy (i.e., the F_1-score), for which the original sentence selection strategy outperforms all alternative strategies with p-value < 0.05.

The answer to RQ4 is that the sentence selection strategy used by FITI outperforms other alternative strategies. Such a strategy, based on phrase lists, improves the precision of sentence selection.

4.6 FITI for Financial Document Compliance Checking (RQ5)

To answer RQ5, we simulate the scenario of compliance checking for financial content requirements and analyze the accuracy of FITI in the context of such scenario.

4.6.1 Methodology

In our experiments, we assessed the accuracy of FITI in identifying information types with a set of approved prospectuses, which satisfy all content requirements. However, in actual compliance checking cases, regulator agents usually inspect (unapproved) prospectuses that do not fulfill some content requirement (i.e., they lack some specific information). We simulated this scenario to understand how FITI can help regulator agents check financial documents for compliance.

Since unapproved prospectuses are usually unavailable due to confidentiality reasons, we simulated such prospectuses by removing sentences related to an information type. Specifically, we first sampled 10 documents from the 50 testing documents as unapproved prospectuses, to simulate the case where a small subset of documents is incomplete. Second, we removed sentences related to the five information types, producing five sets of unapproved documents. Each of these sets contains documents without the sentences related to a given information type. However, when explaining the core content of an information type, submitters may also write additional sentences near the related sentences to introduce the context or background. We assume that for missing information types, submitters would also omit such sentences. Therefore, in all the documents, we removed both the previous sentence and the next sentence of a sentence related to an information type (i.e., the context of a sentence as explained in Section 3.2). Additionally, we manually checked the documents to remove sentences that indirectly indicate the existence of a given information type. Such sentences are usually headings in the table of contents or pointers to specific sections that mention the name of the information type. For example, for *issue condition and procedure* (T5), a prospectus may contain the sentence "shareholders should consult the Chapter *How to Subscribe For Shares*", which determines the position

Table 7 Accuracy of FITI in Detecting Missing Information Types

ID	Precision	Recall	F_1-score
T1	0.70	0.70	0.70
T2	0.75	0.90	0.82
T3	0.90	0.90	0.90
T4	1.00	0.90	0.95
T5	0.83	0.50	0.63
Avg.	0.84	0.78	0.80

of T5 in the text. Lastly, we built a new test set containing 50 testing documents, ten of which are incomplete in terms of information types.

We ran FITI on the new test set using the same cross-validation setting as in RQ1. When FITI reports that it did not find any sentence related to some information types, regulators are warned of missing information types. We defined $Precision = \frac{|TP|}{|TP|+|FP|}$ and $Recall = \frac{|TP|}{|TP|+|FN|}$, where TP means FITI correctly identifies a missing information type (e.g., no related sentence is recommended); FP represents the case in which FITI reports a missing information type but the document contains some related sentences; FN corresponds to the case in which FITI recommends sentences for a missing information type. We also calculate F_1-*score* according to precision and recall. The definitions of evaluation metrics for compliance checking are different from those used in RQ1. In this RQ, we assess the accuracy of FITI on finding information types instead of finding concrete sentences. We focus on two cases, namely FITI cannot find any sentence or find at least one sentence related to an information type.

4.6.2 Results

As shown in Table 7, FITI achieves a precision ranging from 0.70 to 1.00 and a recall from 0.50 to 0.90 when detecting missing information types on the new test set. For T1 to T4, FITI identifies between 70% and 90% of the missing information types. In contrast, the recall value of T5 is low. We speculate that this is caused by the high number of sentences (on average, 36.4) in the training set that are related to T5: many of the annotated sentences may discuss the general background of T5 instead of its core investment information. When using such general sentences to analyze a new prospectus, FITI could wrongly consider similar sentences in other locations as related; hence no warning of missing information types is reported. Regarding precision, on average 84% of the reported missing information types are correct.

The answer to RQ5 is that FITI detects 78% of the missing information types with an average precision value of 0.84, which is a first significant step towards the semi-automated compliance checking of financial documents.

4.7 Threats to Validity

One threat relates to the generality of the study. To address this threat, we chose the content requirements for UCITS prospectuses as it is a representative investment regulatory framework that has managed over €10 trillion of assets across the world for the past 30 years. We evaluated FITI with 100 prospectuses from different investment companies. These prospectuses are representative since they have been published over many years and show a large diversity both in terms of number of pages and number of sentences. The difference in writing styles and document structures demonstrates the accuracy of FITI even in the context of widely varying financial documents. Further, we selected a subset of five representative information types due to the time required to annotate documents and build the training set. To increase generality, we selected these information types by considering their importance, complexity, and diversity. These information types are diverse in terms of the average number of related sentences. They are also diverse in terms of the wording and writing styles as explained in Section 4.1. Therefore, FITI is expected to behave the same way with other requirements that have similar levels of details (i.e., from a single sentence to several pages of sentences) as the five information types considered in this work. Using FITI for tracing other information types requires domain experts to annotate the related sentences for the information types in a set of documents that will constitute the training set of the ML model, and collect the key-phrase lists during the annotation.

Another threat relates to the process of creating the dataset. The annotation of information types is a subjective process. Hence, the annotation was conducted by three domain experts. They annotated the ground truth independently and discussed possible inconsistencies to mitigate the subjectivity from a single domain expert. Meanwhile, to answer RQ4, given the unavailability of unapproved prospectuses due to confidentiality reasons, we simulated unapproved prospectuses by removing related sentences and

any sentence indicating the existence of information types. The documents resulting from such a process may be different from actual, unapproved prospectuses, since the causes leading to a failed approval of a prospectus could manifest in several ways (e.g., with the omission of larger blocks of texts instead of only of the related sentences). Such potential differences could affect the accuracy of FITI when identifying missing information types. In the future, we plan to use actual unapproved prospectuses to evaluate FITI.

Third, the effectiveness of FITI may be affected by the presence of nuanced sentences in financial documents; this implies the meaning of a sentence could change when only a specific word or the context is different. To better analyze the meaning of different sentences, FITI mitigates this threat in two ways. First, as described in Section 3.3, FITI uses word importance similarity to identify the important words. By analyzing the frequency, specificity, and universality of words, FITI is expected to identify some unique words for a given information type. Second, FITI integrates the domain knowledge of experts with two phrase lists (i.e., *rlist* and *ulist*). These lists contain the keywords used by domain experts to search the possible location of related sentences; they are expected to help FITI identify the differences between sentences.

Finally, since regulations can change over time, it could be that the language of the prospectuses in our dataset could change significantly depending on the issue date of the documents, affecting the performance of FITI. We mitigated this threat by assembling an experimental dataset that is not affected by the changes in the law. More specifically, all the information types selected by the domain experts for this study have been required since the establishment of the UCIST Law in 2010, and have not been affected by law changes since then.

5 Discussion

Inspection of financial documents.

FITI can help regulator agents manually inspect financial documents.

Given a new financial document, FITI can analyze it and warn regulator agents of possibly missing information types. According to the results in RQ5, on average FITI can find 78% missing information types with a precision value of 84%. Regulator agents could then browse the document to confirm the warnings.

Besides, FITI can also help regulator agents manually investigate financial documents. Given a financial document, FITI can find 64.6% sentences related to an information type with an average precision value of 82.4% as shown in RQ1. When FITI suggests these related sentences, regulators can quickly read this small set of sentences (usually contains less than 50 sentences, depending on the average number of related sentences in labeled documents) and check more carefully the sentences related to information types to refine their analysis. Thanks to its high precision, in most cases FITI can help regulator agents locate the right position of the sentences related to a certain information type.

As part of future work, we plan to conduct a user study to analyze the effect of FITI on reducing the inspection time of financial documents.

Change of regulations.

The change of the national and international laws can affect FITI, since the content requirements of prospectuses might change substantially over time due to changes introduced in the law: the model trained on labeled prospectuses may not correctly identify the information types written according to the new laws. This problem can be solved by re-annotating the prospectuses with the information types affected by the new laws and then by retraining the model.

We remark that changes in the law are infrequent [16]. For example, all the information types (i.e., T1–T5) we assess in this study are required to be present in every prospectus since the establishment of the "UCITS Law 2010". Hence, the document re-annotation and model re-training steps are not expected to occur frequently.

Moreover, since our evaluation has shown that FITI can identify information types accurately with a relatively small number of labeled documents, the re-annotation step resulting from changes in the law may not be such an impractical undertaking.

Large language models.

In this work, we have not considered large language models (LLM) [12, 66] for two reasons.

First, although there exist a number of LLM-based open-source and commercial solutions (e.g., ChatPDF[4], PDFChat[5], PDF2GPT[6], PDFGPT[7]) that allow a user to upload a PDF and ask questions about its content, all[8] of them ultimately rely on 3rd-party services (e.g., OpenAI API[9]) for processing the content of PDF files and retrieving answers to questions. Since financial documents like prospectuses are to be treated as confidential documents at the time a regulator performs compliance checking (i.e., before the associated fund and the corresponding documents are made available to the public), the use of LLM-based services could raise confidentiality issues until such services could be fully run locally.

Second, LLMs are not specifically trained for tracing content requirements of financial documents. Specializing LLMs for this task and for the financial domain [45] requires tremendous computation resources and would represent a major research endeavor by itself. Recently, some LLMs specialized for the financial domain (such as CFGPT [42] and BloombergGPT [65]) have been released. However, these models have been evaluated only in terms of sentiment analysis, named entity recognition, and summarization tasks; assessing their performance on other tasks such as tracing content requirements of financial documents and developing dedicated prompt engineering best practices are open problems.

6 Related Work

Our approach is related to work done in the areas of requirement traceability, mining financial data, information extraction from regulatory documents, and information structure identification.

6.1 Requirements traceability

Requirements traceability can be considered as a kind of information extraction [29], which extracts entities (e.g., named entities [41]), relations (the relationship between two entities [37]), and events (e.g., knowledge about incidents [36]) from the text with either rule-based [54] or statistical [41] techniques. However, these tasks usually focus on analyzing small pieces of text (e.g., conversations, newsgroups, and weblogs) [52, 64, 71]. Driven by the recent deep learning advances (such as BERT [23] and SpanBERT [38]), the effectiveness of information extraction for analyzing complex documents has also significantly improved. For example, Chalkidis et al [16] performed document-level analysis with a BERT model to extract regulation documents of companies that are affected by a certain law. BERT is also widely used to build legal information retrieval systems [57] However, applying these techniques to a different area (such as the financial domain) always requires some sort of fine-tuning with domain-specific datasets. Since fine-tuning is sometimes unstable on small datasets (with less than 10k training samples) [70], these techniques cannot be applied in our context. We indeed need to account for the limited size of domain-specific datasets, bounded by the high cost of annotations, which must be performed by domain experts and cannot be, for example, crowd-sourced. Moreover, the work by Chalkidis et al [16] aims to retrieve a set of documents that are relevant to a specific document (e.g., an EU directive) from a pool of documents (e.g., national laws) and assumes to have a mapping (i.e., a transposition relation) between EU directives and national laws to define relevance for the retrieval task, whereas FITI works at the sentence-level and does not require any mapping between regulations and prospectuses. Castano et al [14] retrieve legal documents by building the legal ontology. Their approach progressively enriches the terminological knowledge related to a concept and uses the enriched terms to retrieve documents. However, the legal expert is required to review the new terms, while FITI is an automated approach.

Requirements traceability has also been studied in the area of software engineering [2, 18], where many requirement artifacts are written in NL [9, 47, 62]. Existing work infers trace links between high-level NL requirements (e.g., regulatory code) and low-level NL requirements (e.g., requirement specifications and privacy policies). The traceability task is usually recast into an IR problem: taking high- or low-level requirements as queries to retrieve related or similar sentences from low-level requirements [33]. IR techniques including latent semantic indexing, thesaurus, and relevance feedback have been investigated for this task [34]. To address the term mismatch between high- and low-level requirements, the domain ontology [30], word embedding [61], and indicator term

[4] https://www.chatpdf.com
[5] https://www.pdfgpt.chat
[6] https://pdf2gpt.com
[7] https://github.com/bhaskatripathi/pdfGPT
[8] At the time of writing this article, the PDFGPT GitHub page reported an upcoming release with support for LLMs that could be run locally, such as Falcon, Vicuna, Meta LLaMA.
[9] https://openai.com/product

mining [17, 62] methods have been explored for better sentence matching. In addition, for a certain type of artifact (e.g., privacy policies), the NL text can be visualized [51] or standardized with domain-specific languages [13] to improve its traceability.

This study focuses on the traceability of content requirements, a type of non-functional requirements. The concept of "content requirement" was initially proposed in the development of content-intensive interactive applications [7, 8] (e.g., Web sites). For example, in the case of a museum Web site, content requirements might be: "present details for each painting" or "present museum collection history" [7]. This concept has also been recently investigated by Ceci et al [15], who defined content requirements found in the law as "*deontic rules* requiring that some information is contained within an official document"; this is the definition of content requirement that we have considered in this work. Although Ceci et al [15] took financial regulations as a case study to define a model for content requirements, they only discussed its potential application to support compliance checking; they did not provide an approach to automatically identify content requirements for compliance checking. In this work, we take financial documents as a case study to automatically trace content requirements.

Regarding information type identification in content requirements, several studies focus on the analysis of privacy policies. In different countries, privacy policies are subject to compliance with the law (e.g., the General Data Protection Regulation (GDPR) in Europe). For example, GDPR specifies that the privacy policies should indicate "*from which source personal data originates, and if applicable, whether it came from publicly accessible sources*" (Art. 14.2(f)). Torre et al [61] proposed an AI-assisted approach to trace the sentences related to these information types in privacy policies. Amaral et al [2] improved this approach to enable automatic compliance checking between privacy policies and GDPR. In the aforementioned works, a typical requirement is usually one or two sentences in length [34]; in contrast, in this work we have focused on complex NL artifacts (i.e., financial documents), which have thousands of sentences. Further, similar sentences may have different meanings when the context differs; this is not the case for many artifacts (e.g., privacy policies). To address these unique challenges, we proposed FITI to fully consider the context, the content, and indicator words of each sentence for better tracing financial content requirements.

6.2 Mining financial data

Collecting financial data (e.g., financial news, annual financial reports) plays a critical role in improving the quality of financial services and minimizing the risks of financial activities (e.g., portfolio selection [72], stock trading strategy analysis [48], stock price movements prediction [26, 59]). Existing studies report that financial data from Web media (e.g., financial news and discussion boards) has become increasingly salient for analyzing stock markets [44]. Arslan et al [4] and Fan et al [25] cluster and classify financial news to help analysts capture the core events in news.

Data mining has been applied not only to financial data from Web media, but also to financial documents (e.g., annual financial reports, 10-K). Li et al [43] extract financial tables from annual financial reports, and automatically classify them into income statements, balance sheets, and cash flow. Mining financial tables has also enabled activities like financial data cross-checking [40], key performance indicators tracing [10], and financial fraud detection [19, 53].

As to the analysis of NL sentences in financial documents, Kumar et al [39] present a system AEFDT to identify financial named entities (e.g., amortization expense, swing factor). Azzopardi et al [5] propose a controlled NL to write financial statements for financial service compliance checking. In contrast to these works, in this paper we focus on the task of tracing content requirements in financial documents, which is important for financial enterprises and regulators to ensure the completeness of financial documents and further enable automated compliance techniques.

6.3 Information Extraction from Regulatory Documents

In the domain of building construction, several techniques have been proposed to extract information from construction regulatory documents, which are typically large documents that specify construction regulations. The requirements in these documents can be classified into quantitative requirements and existential requirements. Quantitative requirements define the relationship between an attribute of a certain building element/part and a specific quantity value. For example, "Habitable rooms shall have a net floor area of not less than 70 square feet". Existential requirements require the existence of certain building elements/-parts. For example, "The unit (efficiency dwelling unit)

shall be provided with a separate bathroom". Existing approaches extract and organize these regulatory sentences into a computer-processable rule representation (e.g., a structural tuple ⟨Subject, Attribute, Value⟩) for compliance checking with actual building designs. Zhang and El-Gohary [68] [67] designed a set of pattern-matching-based rules to match each sentence in construction regulatory documents for extracting structural tuples. Zhang and El-Gohary [69] used deep learning and transfer learning to extract semantic and syntactic information elements from building regulations.

In building construction, information elements to be extracted are usually words or phrases (e.g., subject and value). Existing studies [67] aim to extract elements of a structural tuples from sentences, and organize them as tuples for analysis. In contrast, FITI analyzes the content requirements in financial documents. Each content requirement can be related to diverse numbers of sentences (e.g., from 3 to 36 sentences).

6.4 Information Structure Identification

In scientific articles, content requirement analysis can be considered similar to the task of *information structure (IS) identification*, which determines the topic or focus of a sentence in a given context [50]. A typical application of IS identification is to analyze the information types of sentences in scientific articles. Guo et al [31] annotated sentences in abstracts of scientific articles with seven categories (background, objective, method, result, conclusion, related work, and future work); they found that it is much faster for readers to understand IS-annotated articles than unannotated ones. Since scientific articles are always required to including these categories of content, it is important to identify IS automatically. Most works use feature-based machine learning, such as SVMs and logistic regression [11] for this purpose. Various linguistic features and learning strategies have been explored, including sentence similarity, adjacency, part-of-speech, and topics [32, 58]. Using the output of IS identification, Huang and Chen [35] developed a scientific writing advisor, which helps refine scientific articles by suggesting similar sentences in the same IS category.

FITI is different from IS identification approaches for three reasons. First, FITI identifies information types in financial documents. Features for existing IS identification tasks (e.g., sentiment indicators) cannot be applied. Second, in financial documents, only a small number of sentences is related to an information type, which makes it more difficult to identify when compared with other types of text (e.g., those found in sections of scientific articles). Third, FITI aims to check the content requirement completeness of financial documents with respect to the identified information types; as discussed above, existing IS identification works have different objectives.

7 Conclusion

In this paper, we proposed FITI, an approach to automatically identify content requirements in financial documents. Our approach combines IR and ML, to conduct analysis at multiple levels of granularity on financial documents in order to understand the context, semantics, and indicator terms of every sentence. Furthermore, FITI uses a heuristic-based sentence selector, which considers the information from IR, ML, and domain-specific phrases to trace text spans related to the information types specified in the content requirements. We evaluated FITI by assessing its effectiveness in identifying information types based on 100 financial documents from different investment companies. Evaluation results show that FITI can accurately retrieve a large percentage of sentences related to information types, with an average precision value of 0.824. FITI can thus effectively inform regulators about potentially missing information types and assist them in inspecting financial documents.

As part of future work, we plan to improve the performance of FITI in the problematic cases identified in Section 4.2 and to investigate the applicability of FITI on other types of financial documents with different information types. We also plan to conduct a user study to assess the effectiveness of FITI to support the compliance checking of real-world unapproved prospectuses.

Acknowledgement

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 authors have 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 Research

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).

Declarations

- Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- Availability of data and materials: The non-annotated prospectuses, the raw output of FITI for each prospectus file in the test set, and the experiment results are available at https://figshare.com/articles/dataset/FITI_Experiment_Results/24481498. The machine learning models and the annotated prospectuses cannot be distributed due to confidentiality and intellectual property agreements.

References

[1] Abreu C, Cardoso H, Oliveira E (2019) Findse@fintoc-2019 shared task. In: Proc. Financial Narrative Processing Workshop (FNP), pp 69–73

[2] Amaral O, Abualhaija S, Briand L (2023) Ml-based compliance verification of data processing agreements against gdpr. In: 2023 IEEE 31st International Requirements Engineering Conference (RE), IEEE, pp 53–64

[3] Apache PDFBox project (2024) Pdfbox. https://pdfbox.apache.org

[4] Arslan Y, Allix K, Veiber L, et al (2021) A comparison of pre-trained language models for multi-class text classification in the financial domain. In: WWW'21 Companion, pp 260–268

[5] Azzopardi S, Colombo C, Pace GJ (2018) A controlled natural language for financial services compliance checking. In: Controlled Natural Language. MIT, p 11–20

[6] Boella G, Caro LD, Humphreys L, et al (2016) Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. Artificial Intelligence and Law 24(3):245–283

[7] Bolchini D, Paolini P (2004) Goal-driven requirements analysis for hypermedia-intensive web applications. Requirements Engineering 9:85–103

[8] Bolchini D, Garzotto F, Paolini P (2007) Branding and communication goals for content-intensive interactive applications. In: 15th IEEE International Requirements Engineering Conference (RE 2007), IEEE, pp 173–182

[9] Boutkova E, Houdek F (2011) Semi-automatic identification of features in requirement specifications. In: RE'11, IEEE, pp 313–318

[10] Brito E, Sifa R, Bauckhage C, et al (2019) A hybrid ai tool to extract key performance indicators from financial reports for benchmarking. In: DocEng'19, pp 1–4

[11] de Britto FA, Ferreira TC, Nunes LP, et al (2021) Comparing supervised machine learning techniques for genre analysis in software engineering research articles. In: Proc. Int'l Conf. on Recent Advances in Natural Language Processing (RANLP 2021), pp 63–72

[12] Brown T, Mann B, Ryder N, et al (2020) Language models are few-shot learners. Advances in neural information processing systems 33:1877–1901

[13] Caramujo J, da Silva AR, Monfared S, et al (2019) Rsl-il4privacy: a domain-specific language for the rigorous specification of privacy policies. Requirements Engineering 24(1):1–26

[14] Castano S, Falduti M, Ferrara A, et al (2022) A knowledge-centered framework for exploration and retrieval of legal documents. Information Systems 106:101842

[15] Ceci M, Bianculli D, Briand LC (2024) Defining a model for content requirements from the law: An experience report. In: 32nd IEEE International Requirements Engineering Conference, RE 2024, Reykjavik, Iceland, June 24-28, 2024. IEEE, pp 18–30

[16] Chalkidis I, Fergadiotis M, Manginas N, et al (2021) Regulatory compliance through Doc2Doc information retrieval: A case study in EU/UK

legislation where text similarity has limitations. In: ACL'21, pp 3498–3511

[17] Cleland-Huang J, Czauderna A, Gibiec M, et al (2010) A machine learning approach for tracing regulatory codes to product specific requirements. In: ICSE'10, pp 155–164

[18] Cleland-Huang J, Gotel OC, Huffman Hayes J, et al (2014) Software traceability: trends and future directions. In: FOSE'14. ACM, p 55–69

[19] Craja P, Kim A, Lessmann S (2020) Deep learning for detecting financial statement fraud. Decision Support Systems 139:113421

[20] CSSF (The Commission de Surveillance du Secteur Financier) (2010) The Luxembourg law of 17 December 2010 relating to undertakings for collective investment. https://www.cssf.lu/wp-content/uploads/L_171210_UCI.pdf

[21] Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research 7:1–30

[22] Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805

[23] Devlin J, Chang MW, Lee K, et al (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT'19, pp 4171–4186

[24] Doyle D (2024) Default english stopwords. https://www.ranks.nl/stopwords

[25] Fan M, Cheng D, Yang F, et al (2020) Fusing global domain information and local semantic information to classify financial documents. In: CIKM'20, pp 2413–2420

[26] Feng F, Li M, Luo C, et al (2021) Hybrid learning to rank for financial event ranking. In: SIGIR'21, pp 233–243

[27] Financial Times (2024) European ucits fund assets reach €10 tn landmark. https://www.ft.com/content/9551561c-0b60-3c2f-bb06-1610983d4ee5

[28] Friedman J, Hastie T, Tibshirani R, et al (2008) The elements of statistical learning, 2nd edn., Springer Series in Statistics, New York, pp 587–588

[29] Grishman R (2019) Twenty-five years of information extraction. Natural Language Engineering 25(6):677–692

[30] Guo J, Gibiec M, Cleland-Huang J (2017) Tackling the term-mismatch problem in automated trace retrieval. Empirical Software Engineering 22(3):1103–1142

[31] Guo Y, Korhonen A, Poibeau T (2011) A weakly-supervised approach to argumentative zoning of scientific documents. In: Empirical Methods in Natural language Processing (EMNLP). ACL, pp 273–283

[32] Guo Y, Reichart R, Korhonen A (2015) Unsupervised declarative knowledge induction for constraint-based learning of information structure in scientific documents. Trans of the Association for Computational Linguistics (TACL) 3:131–143

[33] Hayes JH, Dekhtyar A, Osborne J (2003) Improving requirements tracing via information retrieval. In: RE'03, IEEE, pp 138–147

[34] Hayes JH, Dekhtyar A, Sundaram SK (2006) Advancing candidate link generation for requirements tracing: The study of methods. Trans on Software Engineering 32(1):4

[35] Huang HH, Chen HH (2017) Disa: A scientific writing advisor with deep information structure analysis. In: Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI), pp 5229–5231

[36] Huang YJ, Lu J, Kurohashi S, et al (2019) Improving event coreference resolution by learning argument compatibility from unlabeled data. In: NAACL HLT'19, pp 785–795

[37] Jiang J, Zhai C (2007) A systematic exploration of the feature space for relation extraction. In: NAACL HLT'07, pp 113–120

[38] Joshi M, Chen D, Liu Y, et al (2020) Spanbert: Improving pre-training by representing and

predicting spans. Trans of the Association for Computational Linguistics 8:64–77

[39] Kumar A, Alam H, Werner T, et al (2016) Experiments in candidate phrase selection for financial named entity extraction-a demo. In: COLING'16 System Demonstrations, pp 45–48

[40] Li H, Yang Q, Cao Y, et al (2020) Cracking tabular presentation diversity for automatic cross-checking over numerical facts. In: SIGKDD'20, pp 2599–2607

[41] Li J, Sun A, Han J, et al (2020) A survey on deep learning for named entity recognition. Trans on Knowledge and Data Engineering

[42] Li J, Bian Y, Wang G, et al (2023) Cfgpt: Chinese financial assistant with large language model. 2309.10654

[43] Li Q, Shah S, Fang R (2016) Table classification using both structure and content information: A case study of financial documents. In: Big Data'16, IEEE, pp 1778–1783

[44] Li Q, Chen Y, Wang J, et al (2017) Web media and stock markets: A survey and future directions from a big data perspective. Trans on Knowledge and Data Engineering 30(2):381–399

[45] Li Y, Wang S, Ding H, et al (2023) Large language models in finance: A survey. In: Proceedings of the Fourth ACM International Conference on AI in Finance. Association for Computing Machinery, New York, NY, USA, ICAIF '23, p 374–382, https://doi.org/10.1145/3604237.3626869, URL https://doi.org/10.1145/3604237.3626869

[46] Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge university press

[47] Merten T, Falis M, Hübner P, et al (2016) Software feature request detection in issue tracking systems. In: RE'16, IEEE, pp 166–175

[48] Nuij W, Milea V, Hogenboom F, et al (2013) An automated framework for incorporating news into stock trading strategies. Trans on Knowledge and Data Engineering 26(4):823–835

[49] Porter M (2024) The porter stemming algorithm. https://tartarus.org/martin/PorterStemmer

[50] Postolache O, Kruijff-Korbayová I, Kruijff GJM (2005) Data-driven approaches for information structure identification. In: Proc. Human Language Tech. Conf. and Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP), pp 9–16

[51] Pullonen P, Tom J, Matulevičius R, et al (2019) Privacy-enhanced bpmn: Enabling data privacy analysis in business processes models. Software and Systems Modeling 18(6):3235–3264

[52] Rajpurkar P, Jia R, Liang P (2018) Know what you don't know: Unanswerable questions for squad. In: ACL'18, pp 784–789

[53] Ravisankar P, Ravi V, Rao GR, et al (2011) Detection of financial statement fraud and feature selection using data mining techniques. Decision Support Systems 50(2):491–500

[54] Robaldo L, Caselli T, Russo I, et al (2011) From italian text to timeml document via dependency parsing. In: CICLing'11, Springer, pp 177–187

[55] Rogers A, Kovaleva O, Rumshisky A (2020) A primer in bertology: What we know about how bert works. Transactions of the Association for Computational Linguistics 8:842–866

[56] Sannier N, Adedjouma M, Sabetzadeh M, et al (2017) Legal markup generation in the large: An experience report. In: Proc. Int'l Requirements Eng. Conf. (RE), IEEE, pp 302–311

[57] Sansone C, Sperlí G (2022) Legal information retrieval systems: State-of-the-art and open issues. Information Systems 106:101967

[58] Séaghdha DO, Teufel S (2014) Unsupervised learning of rhetorical structure with un-topic models. In: Proc. Int'l Conf. on Computational Linguistics (COLING), pp 2–13

[59] Shi L, Teng Z, Wang L, et al (2018) Deepclue: Visual interpretation of text-based deep stock prediction. Trans on Knowledge and Data Engineering 31(6):1094–1108

[60] Stanford NLP Group (2024) Stanford nlp core. http://stanfordnlp.github.io

[61] Torre D, Abualhaija S, Sabetzadeh M, et al (2020) An ai-assisted approach for checking the completeness of privacy policies against gdpr. In: RE'20, IEEE, pp 136–146

[62] Wang W, Dumont F, Niu N, et al (2020) Detecting software security vulnerabilities via requirements dependency analysis. Trans on Software Engineering

[63] Wang Y, Shi L, Li M, et al (2022) Detecting coreferent entities in natural language requirements. Requirements Engineering 27(3):351–373

[64] Weischedel R, Pradhan S, Ramshaw L, et al (2011) Ontonotes release 4.0. LDC2011T03, Philadelphia, Penn: Linguistic Data Consortium

[65] Wu S, Irsoy O, Lu S, et al (2023) Bloomberggpt: A large language model for finance. 2303.17564

[66] Yang J, Jin H, Tang R, et al (2024) Harnessing the power of llms in practice: A survey on chatgpt and beyond. ACM Trans Knowl Discov Data https://doi.org/10.1145/3649506, URL https://doi.org/10.1145/3649506, just Accepted

[67] Zhang J, El-Gohary N (2011) Automated information extraction from construction-related regulatory documents for automated compliance checking. In: Proceedings of the 2011 CIB World Congress

[68] Zhang J, El-Gohary N (2016) Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking. Journal of Computing in Civil Engineering 30(2):04015014

[69] Zhang R, El-Gohary N (2021) A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. Automation in Construction 132:103834

[70] Zhang T, Wu F, Katiyar A, et al (2021) Revisiting few-sample bert fine-tuning. In: ICLR'21, to appear

[71] Zhang Y, Zhong V, Chen D, et al (2017) Position-aware attention and supervised data improve slot filling. In: EMNLP'17, pp 35–45

[72] Zhang Y, Zhao P, Li B, et al (2020) Cost-sensitive portfolio selection via deep reinforcement learning. Trans on Knowledge and Data Engineering