Machine Learning with Applications 24 (2026) 100911



Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa





SNOWTEC: Synthetic Natural language Oversampling With Transformer-based information ExtraCtion for automated compliance checking

Hansi Hettiarachchi [a,*], Mohamed Medhat Gaber [b], Pouyan Parsafard [c], Edlira Vakaj [c]

[a] School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK
[b] Centre for Accident Research and Road Safety — Queensland (CARS-Q), Queensland University of Technology, Brisbane, QLD 4000, Australia
[c] College of Computing, Birmingham City University, Birmingham, B4 7BD, UK

ARTICLE INFO

Dataset link: https://github.com/Accord-Project/CODE-ACCORD/

Keywords:
Automated compliance checking
Information extraction
Entity-relation knowledge graphs
Transformers
Deep learning
Data augmentation

ABSTRACT

The optimisation of Automated Compliance Checking (ACC) in Architecture, Engineering, and Construction (AEC) necessitates the interpretation of building codes into machine-processable formats. As these codes primarily exist in textual form, Information Extraction (IE) became integral to decoding this data, encouraging various IE techniques spanning manual, rule-based, and machine-learning methodologies. Recent research has shown promise in adopting deep learning; however, as far as we know, within AEC, the transformers/language models' potential remains untapped/unexplored, yet they hold state-of-the-art performance across various text-based tasks. To address this gap, we propose an approach based on Synthetic Natural language Oversampling With Transformer-based information ExtraCtion (SNOWTEC), designed to extract entities and relations from regulatory text to convert them into machine-processable knowledge graphs. We involve transformer-based architectures and introduce an innovative data oversampling/augmentation approach addressing data scarcity, which impedes model performance. Our experiments across multiple sub-domains highlight the transformers' strength in identifying relations but also reveal challenges in recognising entities within the AEC domain, providing insights for future research. Data oversampling played a crucial role in improving relation extraction, resulting in a notable 26% average F1 increase.

1. Introduction

Compliance checking is integral within the Architecture, Engineering, and Construction (AEC) sector to ensure the safety, stability, reliability and usability of building designs. However, manual checking approaches are labour-intensive, time-consuming, expensive and error-prone (Fuchs & Amor, 2021). Therefore, Automated Compliance Checking (ACC) has been extensively studied over the past 50 years to improve the accuracy and productivity of the compliance checking processes (Zhang et al., 2023). ACC mainly focused on the interpretation of regulations or building codes into computer-processable/machine-readable formats to facilitate the checks via an intelligent design (Zheng et al., 2024). At present, the building codes are written in textual documents for humans to read. Natural languages are hard to process automatically by computers due to their unstructured nature and inherent complexities, which arose from human-centred design (Manning et al., 2008). These challenges are further amplified in the regulatory texts by their complex structures, ambiguities and domain-specific characteristics. Thus, in enabling ACC, the extraction of information from text emerged as a pivotal and complex phase (Costa et al., 2024; Zhou et al., 2022).

Information Extraction (IE) is a sub-domain of Natural Language Processing (NLP) which aims at retrieving factual information described in free text (Piskorski & Yangarber, 2013). There are two main types of information found in the text: (1) entities (also known as named entities) and (2) relations, which are crucial for understanding the ideas expressed in natural language (Jurafsky & Martin, 2009). An entity is a concept or a specific piece of information that can be categorised, or simply, anything that can be identified by using a proper name (Jurafsky & Martin, 2009). For instance, given the sentence *'The gradient should not exceed five per cent.'*, *'gradient'* and *'five per cent'* represent its entities. A relation is a semantic connection/association among entities in the text (Jurafsky & Martin, 2009). For example, the sentence mentioned above illustrates a *'less equal'* relation between the

* Corresponding author.
E-mail addresses: h.hettiarachchi@lancaster.ac.uk (H. Hettiarachchi), mohamed.gaber@qut.edu.au (M.M. Gaber), pouyan.parsafard@mail.bcu.ac.uk (P. Parsafard), edlira.vakaj@bcu.ac.uk (E. Vakaj).

https://doi.org/10.1016/j.mlwa.2026.100911
Received 26 November 2025; Received in revised form 5 March 2026; Accepted 24 April 2026
Available online 5 May 2026
2666-8270/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

entities *'gradient'* and *'five per cent'*. Collectively, these entities and their relations capture the rule(s) expressed in natural language.

Previous research has proposed various approaches to extract information from text, aiming to enhance the effectiveness of ACC. In early work, manual methods were popularly used for IE due to text complexities and domain-specific demands (Beach et al., 2015; Hjelseth & Nisbet, 2011). However, the labour-intensive nature of manual approaches and their limitations in efficiently supporting ACC prompted a shift towards automating IE utilising NLP and Machine Learning (ML) techniques. This shift mainly introduced rule-based approaches for IE (Zhang & El-Gohary, 2016, 2017; Zhou & El-Gohary, 2017). Despite their performance, rule-based methods inherently lacked adaptability and flexibility, being heavily reliant on domain-specific characteristics (Zhang & El-Gohary, 2021). Their accuracy highly depended on manually crafted rules, requiring extensive domain expertise. These limitations have encouraged a transition within regulatory IE methods towards ML techniques, leveraging predictive models to capture textual information rather than relying on laborious handcrafted rules.

According to recent ML-based research, adopting deep-learning techniques has emerged as a promising avenue for regulatory IE. Bidirectional Long Short-Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN) architectures have been commonly used in extracting entities and relations from regulatory text due to their proficiency in learning long-term dependencies and contextual features (Wang & El-Gohary, 2021, 2023). However, the approaches proposed within the AEC sector for IE exhibit notable gaps, particularly in light of recent advancements in the NLP domain. In NLP, transformer-based architectures have gained significant recognition, showcasing state-of-the-art performance across various tasks, including IE, covering entity and relation extraction (Plum et al., 2022; Shen, Tan et al., 2023; Shen et al., 2022; Yang, Choi et al., 2023). Overall, transformer models (also known as language models) represent a significant advancement in text processing with their sophisticated ability to capture text's contextual details and transfer pre-trained knowledge to downstream tasks (Devlin et al., 2019; Vaswani et al., 2017). Yet their end-to-end application to regulatory IE in the AEC sector, particularly targeting both entity and relation extraction as a unified pipeline, remains largely unexplored.

Our research addresses this gap by formulating regulatory IE in AEC as a transformer-based, end-to-end extraction problem that jointly supports entity and relation modelling under low-resource conditions. Beyond adopting transformer architectures, the contribution lies in systematically adapting, benchmarking, and evaluating these models within a domain characterised by complex sentence structures, domain-specific terminology, and limited annotated data. To mitigate the critical bottleneck of data scarcity, data augmentation has recently emerged as a popular solution in the NLP domain. It includes methods to oversample training data by generating synthetic data from existing samples (Feng et al., 2021). However, there is a notable absence of data augmentation applications within the AEC sector focusing on textual data. Addressing this limitation, we also propose a novel domain-aware data augmentation strategy specifically designed for relation extraction in regulatory text. Unlike generic augmentation approaches, our method leverages structural properties of regulatory sentences to generate consistent synthetic training samples, which enables improved model robustness.

Furthermore, we introduce a comprehensive IE pipeline that converts regulatory sentences into structured knowledge graphs (KGs) of entities and relations to support automated compliance checking (ACC). While existing ACC approaches often focus on translating regulatory text directly into executable rule languages such as SPARQL (Fuchs et al., 2022; Zheng, Zhou et al., 2022), our work proposes an intermediate, yet descriptive representation that is both machine-processable and human-interpretable. This intermediate KG layer serves two important purposes: (1) it enables modular integration with downstream compliance engines, and (2) it facilitates human-in-the-loop validation, which is essential in high-stakes regulatory contexts. By prioritising interpretability, extensibility, and domain adaptation, our approach contributes not only an application of standard architectures but a structured framework for trustworthy regulatory IE in the AEC sector.

Through our experimental study, we compare the performance of several recent transformer models, including Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), Robustly optimised BERT approach (RoBERTa) (Liu et al., 2019) and A Lite BERT (ALBERT) (Lan et al., 2020), employing diverse learning techniques for IE. Also, we utilise data from multiple domains such as accessibility, fire safety, structural integrity, energy efficiency and more to emphasise the general applicability of the proposed methodology. In summary, the main contributions of this paper are as follows.

1. We propose a novel information extraction pipeline named Synthetic Natural language Oversampling With Transformer-based information ExtraCtion (SNOWTEC) utilising state-of-the-art transformer/language models to automatically extract information from the regulatory text in a domain-independent manner.
2. We introduce a novel data oversampling/augmentation method designed to address data scarcity concerns that impede the predictive capabilities of language models.
3. We reformulate the process of information extraction from the regulatory text as a knowledge graph generation task, introducing a new direction to facilitate effective ACC.
4. We release our implementation as an open-source project[1] to support future research and applications within ACC.

The rest of this paper is organised as follows. Section 2 discusses the prior work, mainly spanning information extraction from building regulatory text and recent trends in natural language processing. Section 3 introduces the proposed approach SNOWTEC, the transformer-based information extraction pipeline powered by synthetic natural language oversampling. Section 4 describes the experimental setup covering data, transformer models, hyper-parameters and evaluation metrics. Following this, Section 5 comprehensively reports the conducted experiments, obtained results and error analyses together with an end-to-end SNOWTEC demonstration. Finally, Section 6 concludes the paper with a discussion and possible future directions.

2. Related work

2.1. Information extraction from building regulatory text

In the AEC sector, regulatory information is mainly conveyed through natural language. Thus, to enable ACC, it becomes pivotal to either extract information from text or convert textual content into machine-readable formats. One of the early methods introduced for this purpose is the Requirement, Applicability, Selection, and Exception (RASE) methodology (Hjelseth & Nisbet, 2011). RASE operates on a semantic basis to transform normative documents into structured rules using boolean logic. It requires manual tagging of textual phrases in documents or independent text blocks into distinct categories and manual extraction of metadata relying on both textual information and domain expertise, to generate rules (Beach et al., 2015). However, recognising the human resource demand associated with manual methods, recent research has shifted its focus towards automating information extraction through NLP techniques to streamline ACC processes effectively (Ding et al., 2022).

Entities and relations are the two fundamental types of information found in the text, which are essential for understanding the ideas

[1] Our implementation is available on https://github.com/Accord-Project/accord-nlp.

expressed in natural language (Jurafsky & Martin, 2009). Thus, prior research showcased two sub-areas under automatic information extraction focusing on entity extraction and relation extraction (Zhang & El-Gohary, 2017; Zhou et al., 2022). Early research within both of these areas predominantly aimed at rule-based methodologies which tend to yield high performance (Zhang & El-Gohary, 2016). However, the development of rules necessitates extensive manual efforts. Thus, recent research indicates a tendency towards machine learning, particularly favouring supervised learning-based approaches, as described below.

2.1.1. Entity extraction

Rule-based approaches were widely used to extract entities from regulatory text within the AEC sector. Specifically, rules that rely on pattern matching were commonly used to identify textual entities (Zhang & El-Gohary, 2016, 2017). These rules were mainly written using syntactic features such as part-of-speech (PoS) tags, phrasal tags (generated by phase structure grammar rules) and gazetteer tags, and semantic features captured using domain-specific ontologies. Zhou and El-Gohary (2017) proposed further enhancements to such rules by defining dependency-based and cascaded patterns considering the complex structures in the regulatory text. Xue and Zhang (2022) also proposed a similar pattern-matching approach to extract entities, but they only used PoS tags as syntactic features. All these approaches were experimented on selected domains such as energy conservation or particular chapters of the International Building Code (IBC), emphasising the domain dependence of rule-based approaches and their limited expandability. Since almost all rule-based extraction methods rely on PoS tags, Xue and Zhang (2020) introduced error-driven transformational rules to improve the existing PoS taggers. Later, they also improved their PoS tagger tailored to building codes using deep learning techniques (Xue & Zhang, 2021). However, writing rules require intensive manual efforts which are time-consuming and error-prone. Also, rule-based approaches exhibit less generalisability across different domains as most of the rules are domain-dependent.

Considering the limitations associated with rule-based entity extraction methods, recent research focused more on supervised learning-based approaches, aiming to reduce manual intervention. The adoption of deep learning techniques has emerged as a promising avenue for entity extraction from regulatory text. This trend also aligns with the evolution seen in various machine learning applications, where deep learning approaches have consistently outperformed traditional methodologies. Among different neural network architectures available, the Bidirectional Long Short-Term Memory (Bi-LSTM) model was popularly used for entity extraction within the AEC sector. LSTMs are more capable of learning long-term dependencies utilising their memory cells than vanilla Recurrent Neural Networks (RNN), and their bidirectional architecture captures both past and future states of the sequence which are important for entity recognition (Hassan & Mahmood, 2018). For instance, Moon et al. (2021) proposed a model utilising Word2Vec embeddings and Bi-LSTM architecture to identify entities in specifications for road construction projects. Zhang and El-Gohary (2021) enhanced this architecture by adding a Conditional Random Field (CRF) layer. Similarly, another research proposed a hybrid model of Bi-LSTM and Convolutional Neural Network (CNN) to recognise entities in fall protection (Wang & El-Gohary, 2021). Deviating from this common trend, transformer-based architectures (i.e. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019)) were also developed to extract entities, following the recent advances in NLP (Schönfelder & König, 2021; Zheng, Lu et al., 2022; Zhou et al., 2022).

2.1.2. Relation extraction

The majority of the available approaches for extracting semantic relations from the regulatory text primarily rely on rule-based methodologies. Similar to the early trends in entity extraction, semantic rules were popularly used to identify relations from building codes (Zhang & El-Gohary, 2016, 2017; Zhou & El-Gohary, 2017). These rules were based on both syntactic (e.g. PoS tags, gazetteer tags, etc.) and semantic (e.g. building ontology concepts and their interconnections) features extracted using various NLP techniques. However, these approaches were mainly limited to capturing quantitative and comparative relations in selected domains such as energy conservation or particular chapters of the IBC due to the high domain dependence of rules. Another prevalent method involved was syntactic parsing to analyse the structure of sentences labelled with entities, aiming to extract underlying relations (Zheng, Zhou et al., 2022; Zhou et al., 2022). This approach also requires tailored context-free grammars (CFGs) that rely heavily on domain-specific characteristics or textual patterns, limiting its expandability. This fact is further emphasised by the closed-domain (i.e. fire protection) experiments reported in previous work.

Considering the domain-specific constraints and intricacies inherent in rule-based methods demanding significant manual labour, there is a tendency to aim at supervised learning-based approaches to extract relations from regulatory text, following the same trend seen with entity extraction. However, the limited applications found in the literature indicate that this trend is still in its early stages within the AEC sector. Wang and El-Gohary (2023) proposed two deep neural network architectures: Attention-based CNN and Entity-aware Attention Bi-LSTM to recognise relations between entities. They used Word2Vec and GloVe embeddings to encode text, and their findings indicate that the CNN-based model performed best emphasising the importance of contextual features in identifying relations. Also, they experiment with multiple relation types such as part-of, located-at and not-over, unlike the rule-based approaches, indicating the expandability of the proposed approach. Yet, these experiments are limited to fall protection requirements.

2.2. Recent trends in natural language processing

Within the NLP domain, there is a clear shift from rule-based methodologies to supervised learning across various applications. The improved text encoding utilising the advanced embedding techniques capable of automatically capturing underlying linguistics stands as a major driving force behind this transformation. Word2Vec (Mikolov et al., 2013) is one of the first prediction-based embedding modelling techniques, which attracted wide attention from the NLP community. Transformer models (also known as language models) were introduced lately with the capability to capture the contextual impact on a word's meaning overcoming some of the major limitations associated with classic word embeddings such as Word2Vec. Transformers are deep neural networks which utilise attention mechanism to capture text's contextual details and long-range dependencies (Vaswani et al., 2017). Following transformers, different model variants such as BERT (Devlin et al., 2019), Robustly optimised BERT approach (RoBERTa) (Liu et al., 2019) and Cross-lingual Language Model-RoBERTa (XLM-R) (Conneau et al., 2020) were proposed with the ability to generate contextual text representations and fine-tune for downstream tasks by transferring the pre-trained knowledge. These capabilities allowed the transformer-based architectures to achieve state-of-the-art results in many complex NLP tasks such as machine translation (Ranasinghe et al., 2020), offensive language detection (Zampieri et al., 2023) and question answering (Nassiri & Akhloufi, 2023).

The entity and relation extraction tasks also exhibit the same trend in utilising transformer-based models within the NLP domain, considering their superior performance. For instance, BERT-based architectures were popularly used to identify common entities in newswire text such as location, organisation and person (Shen, Song et al., 2023; Shen et al., 2022). Shen, Tan et al. (2023) proposed a RoBERTa-based architecture to extract such common entities. BioBERT and PubMedBERT-based architectures were also developed targeting entities in biomedical text (Luo et al., 2022; Shen et al., 2022). Moreover, a recent study explored the performance of different transformer models (e.g. BERT,

XLM-R, etc.) in identifying entities in event-described text, following their cutting-edge performance (Hettiarachchi et al., 2023). Within relation extraction, Plum et al. (2022) proposed a BERT-based architecture to identify biographical relations in text. SpanBERT, a modified version of BERT, was also involved in extracting commonly appeared relations in newswire and web text, recently (Joshi et al., 2020). Luo et al. (2022) proposed using BioBERT and PubMedBERT models to identify relations in biomedical text. This trend is also extended to historical relations by utilising the multilingual BERT model and setting the state-of-the-art performance (Yang, Choi et al., 2023).

2.3. Summary

As detailed in Section 2.1, previous literature reveals the utilisation of diverse NLP techniques to extract information from regulatory texts within the AEC sector. However, these approaches exhibit gaps, particularly in light of recent advancements in the NLP domain. Although transformer-based architectures have been employed for entity identification in building codes recently, existing experiments have predominantly relied on the BERT model, limiting exploration (Zheng, Lu et al., 2022; Zhou et al., 2022). Also, to the best of our knowledge, transformer-based methodologies have not been involved in regulatory relation extraction within the AEC sector, despite their state-of-the-art performance in NLP. Our primary objective is to address these gaps by integrating transformer-based architectures for both entity and relation extraction from regulatory texts in the construction domain. Additionally, we aim to assess the performance of various transformer models employing diverse learning techniques, expanding beyond the application of the BERT model.

The need for large amounts of annotated data to train large-scale neural networks can be mentioned as one of the main reasons that impede the involvement of recent advancements in NLP within the AEC sector. Even though pre-trained transformer models provide initial linguistic knowledge, they also require substantial data to get fine-tuned for downstream tasks successfully. Following this requirement, data augmentation has recently emerged as a popular technique in the NLP domain to address data scarcity issues (Feng et al., 2021). Common text augmentation techniques include back translation, synonym replacement and paraphrasing (Feng et al., 2021; Obadinma et al., 2023; Shanbhogue et al., 2023). However, these methods are primarily suited for general NLP tasks as they may alter domain-specific keywords crucial for tasks like information extraction from the regulatory text. Looking into the AEC sector, there are recent studies which utilised image data augmentations (Yang, del Rey Castillo et al., 2023; Zhai et al., 2022); however, no approaches that utilise text data augmentation are found. In this work, we propose a novel data augmentation method tailored for domain-specific textual data, such as regulatory text and investigate its impact on transformer model performance, aligning with our primary objective to bridge these gaps.

Moreover, our focus extends to developing an information extraction pipeline integrating supervised machine learning models for both entity and relation extraction, which was absent in prior research. This approach aims to streamline information extraction while reducing manual effort and domain-specific requirements, which hinder the expandability. Additionally, unlike most of the existing experimental studies that concentrate on specific chapters or domains, our objective involves analysing the performance of proposed methods across multiple sub domains within the AEC sector. Such an assessment will be pivotal to ascertain the methods' generalisability, which is critical for their real-world applicability.

3. SNOWTEC

Synthetic Natural language Oversampling With Transformer-based information ExtraCtion (SNOWTEC), the information extraction pipeline introduced in this paper comprises four integral components: (1) entity classifier, (2) entity pairer, (3) relation classifier and (4) graph builder. As illustrated in Fig. 1, the architecture follows a sequential processing scheme in which the output of each module serves as the input to the subsequent stage, enabling a structured transformation from unstructured text to a machine-processable knowledge graph.

The pipeline begins with the entity classifier, a transformer-based model that identifies and categorises entities within a raw sentence. Its predictions define the nodes of the resulting graph. The entity pairer then constructs all candidate entity pairs from the detected entities, reformulating the entity classifier output into the structured input required by the relation classifier. Following this step, the relation classifier, also based on a state-of-the-art transformer architecture, determines the semantic relationship between each candidate pair. This stage effectively filters and labels entity interactions, distinguishing meaningful relations from non-relations. Finally, the graph builder integrates the valid entity–relation predictions into a structured graph representation, where entities correspond to nodes and predicted relations to edges.

Together, these modules form a comprehensive IE pipeline in which classification components (entity and relation classifiers) perform semantic extraction, while transformation or connecting components (entity pairer and graph builder) ensure architectural consistency and data flow compatibility. The classification components are further described in Sections 3.2 and 3.3, following an introduction to general transformer architecture in 3.1. The complete workflow is presented in Fig. 1, including an example input, intermediate representations, and the final graph output, to clarify the interaction between modules. Additionally, Algorithm 1 provides an algorithmic overview to the entire process, and Section 3.4 discusses its computational complexity.

Overall, when provided with a regulatory sentence as input, SNOWTEC produces a machine-readable output (i.e. an entity-relation graph) that encapsulates the information expressed in natural language. This graph can be readily processed to automate compliance checks. Also, SNOWTEC relies on supervised machine learning models, demanding fewer human resources compared to methods that depend on domain-specific rules for capturing semantic relationships. Apart from automating relation extraction in a domain-independent manner, we also propose a novel data augmentation method to synthetically generate data to train language models effectively (Section 3.3.1). Moreover, as far as we are aware, this is the first effort to convert a regulatory sentence into a knowledge graph to facilitate ACC.

3.1. Transformer architecture for text encoding

SNOWTEC involves transformer-based architectures to conduct entity and relation extraction from free text, focusing on their effectiveness in many NLP applications (Devlin et al., 2019; Hettiarachchi et al., 2021; Ranasinghe et al., 2020). Even though transformers are not widely used targeting ACC within the AEC sector, they form state-of-the-art entity and relation extraction methods within the NLP domain (Ma et al., 2023; Shen, Tan et al., 2023). Most transformer models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) and Robustly optimised BERT approach (RoBERTa) (Liu et al., 2019) provide stronger results than models based on Recurrent Neural Network (RNN) architectures while providing language models pre-trained on large text corpora facilitating the successful fine-tuning of downstream tasks. Also, transformers consist of multi-layer bidirectional encoders that utilise the self-attention mechanism (Vaswani et al., 2017) to produce language representations which capture underlying linguistics and the context. A transformer encoder processes a sequence of text (e.g. sentence) and outputs representations/embeddings that correspond to the entire sequence and its tokens, which can be used for the learning of downstream tasks while conserving the original text's linguistic features. Fig. 2 represents a general overview of the transformer encoder architecture.

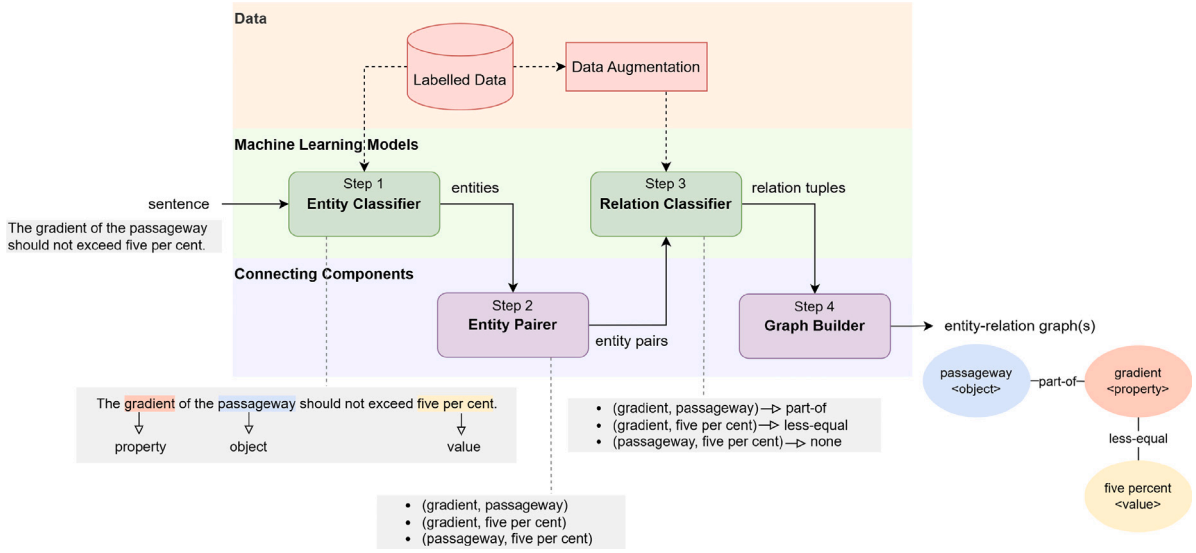


Fig. 1. Overview of the proposed method – SNOWTEC.

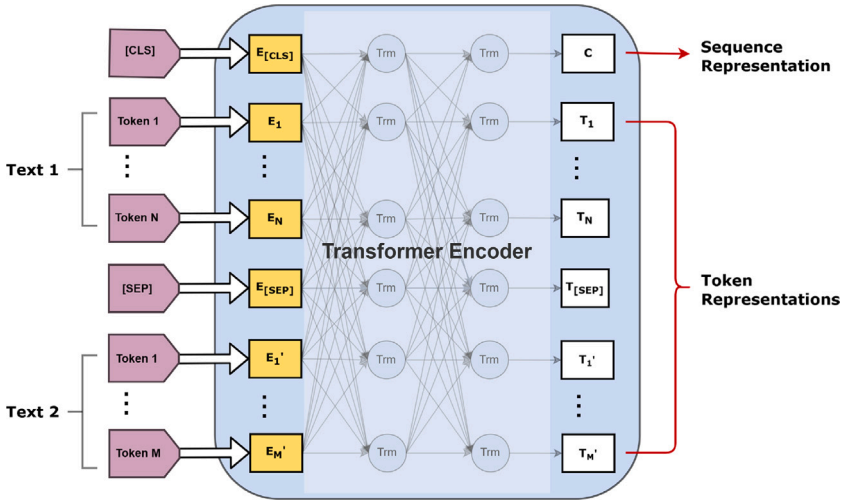


Fig. 2. Conceptual overview of the transformer encoder architecture. The internal layers are abstracted for clarity; each layer corresponds to a standard transformer encoder composed of multi-head self-attention and feed-forward sublayers with residual connections (Vaswani et al., 2017).

Transformer input format:. The transformer encoder takes a text sequence or a text sequence pair as its input, enabling it to handle diverse downstream tasks. It uses special tokens, including [CLS] and [SEP], to organise the input text. [CLS] is the initial token, which indicates the beginning of the sequence. [SEP] is the separator to be placed in between if the input has two sequences. After this raw text formatting, a tokeniser converts the sequence into a token embedding. In addition to the token embedding, a segment and a position embedding are required to format the input. The sum of these three embeddings constitutes the transformer's input. The segment embedding contains boolean values (i.e. 0 and 1) indicating the separation of sequences/segments. The position embedding contains sequential numbers starting from 0, which specify each token's position within the sequence.

Transformer output format:. The transformer encoder generates representations/embeddings per token in the input sequence. The output of the initial token ([CLS]) is an embedding representing the complete input sequence. It can be used for sequence-based predictions/classifications or as a contextual sequence embedding. Similarly, other outputs are token embeddings corresponding to each input token, which form contextual word/token embeddings that can be used with token-based predictions/classifications. The transformer architecture needs to be modified by including an additional layer, like a classification head suitable for the targeted task, on top of the output layer to fine-tune the model for a downstream task.

3.1.1. Learning methods

Transformers undergo two training steps according to their design: (1) pre-training/language modelling and (2) fine-tuning.

Pre-training/language modelling:. In the pre-training phase, a language model is built by training the transformer architecture on unlabelled textual data or free text. Therefore, pre-training is also known as language modelling. There are two commonly used pre-training approaches: (1) masked language modelling (MLM) and (2) next sentence prediction (NSP) (Devlin et al., 2019). The MLM approach masks a portion (e.g. 15%) of the input tokens randomly and then trains the model to predict the masked tokens. This task helps to learn bidirectional representations by training the model to predict masked tokens in a multi-layered context while focusing on both directions. NSP is a binary prediction task that focuses on recognising whether a sentence pair appears consecutively in a monolingual corpus. This task mainly helps

Algorithm 1 Information Extraction Pipeline

Input: S // *a regulatory sentence*
Output: $G = (V, E)$ // *a labelled graph*

// *Step 1: Entity Classification*
$E_{\text{raw}} \leftarrow$ EntityClassifier(S) // *Returns* $\{(e_i, \tau_i) \mid e_i \subseteq S,\ \tau_i \in EntityTypes\}$
$V \leftarrow E_{\text{raw}}$ // *Add graph nodes (entity spans with labels)*

// *Step 2: Entity Pairing*
$P \leftarrow \emptyset$
for each $(e_i, _) \in E_{\text{raw}}$ **do**
  for each $(e_j, _) \in E_{\text{raw}},\ j > i$ **do**
    $p_{ij} \leftarrow (e_i, e_j)$
    $P \leftarrow P \cup \{p_{ij}\}$
  end for
end for

// *Step 3: Relation Classification*
$E \leftarrow \emptyset$
for each $p_{ij} \in P$ **do**
  $r_{ij} \leftarrow$ RelationClassifier(S, p_{ij}) // *Returns* $r_{ij} \in RelationTypes$
  if $r_{ij} \neq none$ **then**
    $E \leftarrow E \cup \{(e_i,\ r_{ij},\ e_j)\}$ // *Add valid graph edge*
  end if
end for

// *Step 4: Graph Construction*
$G \leftarrow$ BuildGraph(V, E)
return G

the model to understand the relationships/interconnections between sentences. Among these two techniques, MLM is commonly used by various transformer architectures such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020), as it was found to be competitive. Thus, we also involve MLM for language modelling within this research.

Fine-tuning:. Fine-tuning is normally conducted by targeting a downstream task. To train a transformer for a downstream task, an appropriate layer(s), such as a classification head, needs to be added to the top of the transformer's output layer, depending on the targeted task. During this process, the model initialises using its pre-trained parameters and then gets fine-tuned for the targeted task utilising the task-specific labelled data. The initialisation with pre-trained parameters commences the learning process with the model's original knowledge and facilitates the effective learning of the downstream task. Fine-tuning can be applied to all model parameters (i.e. parameters of the transformer and its additional layers) or only to the additional layers. According to previous research, fine-tuning the transformer together with its newly-added layers obtained the best results for downstream tasks, as this approach adjusts the weights in the entire architecture, including the generic language model focusing on a particular task or domain-specific data rather than only fine-tuning the task-specific layers (Merchant et al., 2020). Thus, our approach also follows the entire fine-tuning process. More details about the modifications we made to the transformer architecture to support entity and relation classification are described in the below sections.

3.2. Entity classifier

We formulate entity classification as a sequence labelling problem, as all the input tokens can be associated with entity labels. We use the BIO (Beginning, Inside, Outside) format, regarded as the standard for information extraction tasks (Ramshaw & Marcus, 1995), to arrange entity labels to use with the model. Since an entity can span over multiple words/tokens, B marks the beginning word, and I marks the other words which belong to the entity. All the remaining words which do not represent any entity will be marked by O. Following this format, we provide a label for each word in the text sequence, converting the entity recognition task into a sequence labelling problem. A sentence together with its BIO-tagged entities is shown in Fig. 3.

We modify the general transformer architecture shown in Fig. 2 with softmax layers for each output token as illustrated in Fig. 4, following the sequence labelling architecture commonly used with transformers with remarkable results (Devlin et al., 2019; Hettiarachchi et al., 2023). Since the model requires predicting entities in a single sentence per instance, we only use the [CLS] token for input formatting without including the [SEP] token. A softmax layer consists of k number of neurons equal to the class/category count aimed at by the classifier (i.e. entity types). Each neuron adopts the softmax activation function following Eq. (1), which returns P_i, the probability per class i. The input and output vectors are represented by z_i and z_j. Once all probabilities are calculated, the class with the highest probability is identified as the final prediction.

$$P_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{1}$$

3.3. Relation classifier

For relation extraction, we adapt the transformer-based architecture shown in Fig. 5, which was proposed by recent studies in the NLP domain Baldini Soares et al. (2019), Plum et al. (2022). Since the aim is to identify the semantic connection between an entity pair, we involved four additional special tokens: <e1>, </e1>, <e2> and </e2> to format the model's input. <e1> and </e1> denote the start and end of the first entity that appeared in the text sequence from the selected entity pair. Similarly, <e2> and </e2> denote the start and end of the second entity as shown below:

> The <e1>gradient</e1> of the <e2>passageway</e2> should not exceed five per cent.

Among the transformer's special tokens, we only use [CLS] with the relation classifier, as it requires processing a single sentence per instance, similar to the entity classifier. From the output layer, we take the embeddings corresponding to <e1> and <e2> as linguistical representations for the given entities. Then, their concatenation is considered the final output representation, which is fed into a softmax layer to predict the associated relationship. As described in Section 3.2, a softmax layer calculates the probability per class (i.e. relation types), following Eq. (1). We pick the relation type with the highest predicted probability as the final prediction.

Additionally, we propose a novel data augmentation approach for relation data, involving entity data in Section 3.3.1. Deep neural networks typically benefit from improved performance when trained on large datasets, enabling them to learn more effectively.

3.3.1. Data augmentation

Data augmentation is the process that constructs synthetic data from an available dataset. Such a process proves advantageous in scenarios with limited or unbalanced data, mitigating the risk of overfitting in model training (Feng et al., 2021). Even though data augmentation has been widely used with image data, its utilisation in NLP is still at an early stage (Shorten et al., 2021). One of the primary challenges in augmenting textual data lies in preserving the intricate linguistic structure available in original data with synthetic data.

Back-translation and random synonym replacement are the two popularly used data augmentation techniques within NLP, considering their ability to generate data without losing underlying linguistics

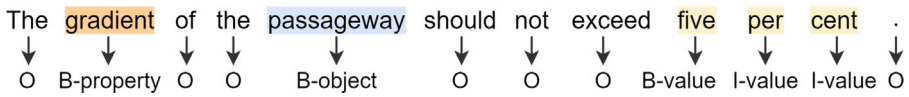


Fig. 3. Entity-labelled sentence in BIO format. This sample was acquired from the CODE-ACCORD dataset, which if further described in Section 4.1.

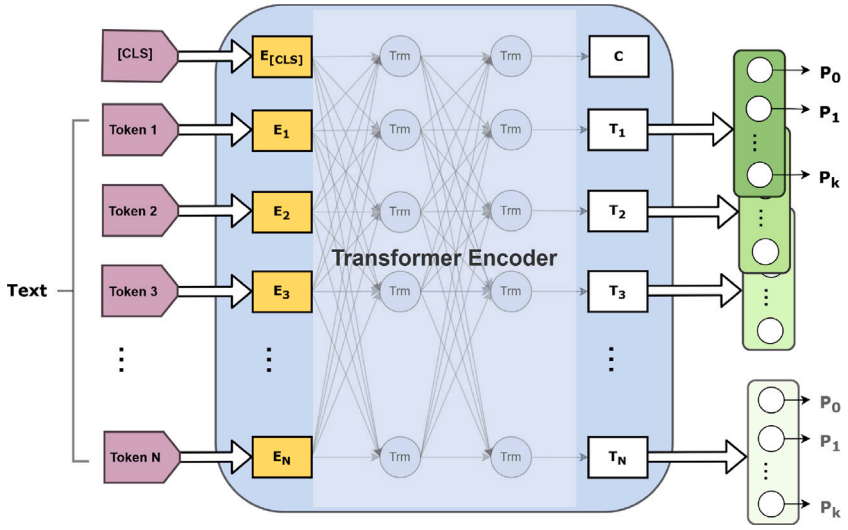


Fig. 4. Entity classification architecture.

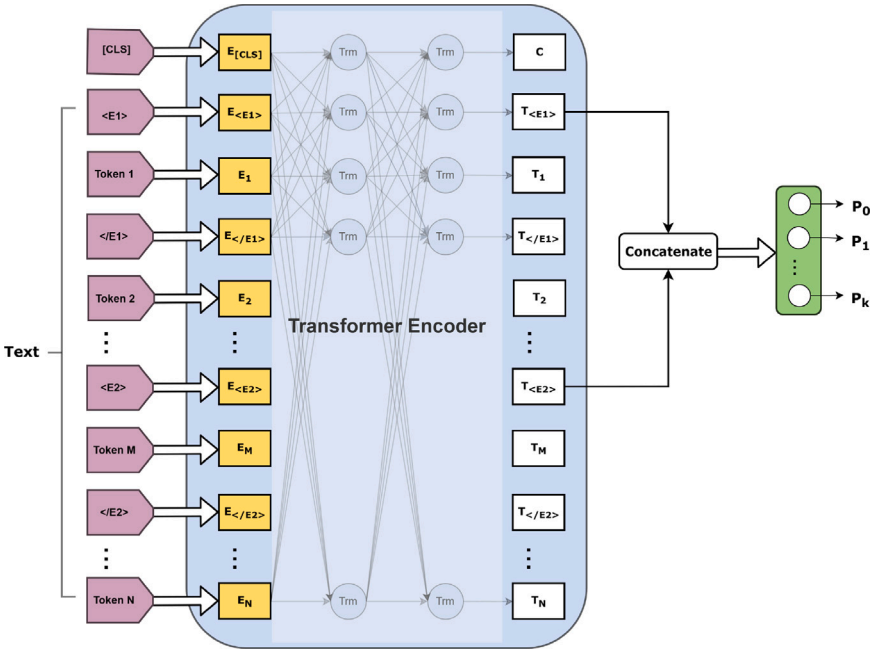


Fig. 5. Relation classification architecture.

(Jin et al., 2020; Longpre et al., 2020). Back-translation translates text in one language to another and then, back to the original language. However, this approach cannot be automatically used when the original data includes token-level labels as back-translated samples could have a different number of tokens with no direct mapping with the original tokens. The other approach replaces one or more words in the original text with corresponding synonyms. Even though it is generally applicable for token-labelled data, it is not applicable for the regulatory data targeted by this research. The majority of the entities in regulatory data appear as multi-word instances (e.g. 'fire door', 'cross-section area', etc.) and replacing some words with their synonym could result in text phrases which are unknown to the domain or no longer entities.

After carefully analysing the limitations in available data augmentation techniques, we propose a novel method to synthetically generate relation-annotated data utilising domain-specific entities. Each relation-annotated instance includes an entity pair marked in a sentence or textual context with corresponding entity categories and an associated relation, as can be seen in Table 1 (columns Original Sample, e1, e2 and Relation). In such a setting, we can create more samples by replacing one or both entities with other entities of the same category. Table 1

Table 1
Original sentence samples and their corresponding synthetic samples. Columns e1 and e2 represent the categories of each entity. A colour scheme representing categories is involved to highlight the entities in original and synthetic samples.

Original sample	e1	e2	Relation	Synthetic samples
Between the locking points for the mortice lock and surface-mounted rim lock, the ⟨e1⟩ distance ⟨/e1⟩ should be ⟨e2⟩ 400–600 mm ⟨/e2⟩.	Property	Value	Equal	Between the locking points for the mortice lock and surface-mounted rim lock, the ⟨e1⟩ cross-sectional area ⟨/e1⟩ should be ⟨e2⟩ 1m3/ ⟨/e2⟩. Between the locking points for the mortice lock and surface-mounted rim lock, the ⟨e1⟩ emission rate ⟨/e1⟩ should be ⟨e2⟩ 25W/m2 ⟨/e2⟩. Between the locking points for the mortice lock and surface-mounted rim lock, the ⟨e1⟩ power input from controls ⟨/e1⟩ should be ⟨e2⟩ 50 ⟨/e2⟩.
⟨e1⟩ Extinguishing routes ⟨/e1⟩ of basement storeys must not be connected to fire and smoke-proof ⟨e2⟩ exits ⟨/e2⟩.	Object	Object	Not-part-of	⟨e1⟩ dwelling units ⟨/e1⟩ of basement storeys must not be connected to fire and smoke-proof ⟨e2⟩ space heating or airconditioning system ⟨/e2⟩. ⟨e1⟩ room thermostat ⟨/e1⟩ of basement storeys must not be connected to fire and smoke-proof ⟨e2⟩ tears ⟨/e2⟩. ⟨e1⟩ kitchens ⟨/e1⟩ of basement storeys must not be connected to fire and smoke-proof ⟨e2⟩ motor ⟨/e2⟩.

contains three augmented samples per original sample to illustrate the idea further. On some occasions, the generated samples could hold false information. For instance, the first synthetic sample in Table 1 states an area (i.e. cross-sectional area) should be equal to a volume measure (i.e. 1 m^3). However, it illustrates the text pattern and semantic relationship between two such entities within the given context accurately, allowing a model to learn associations among entities given a textual context. This fact is also reinforced by the results reported in Section 5. Also, even though we used samples from the CODE-ACCORD dataset detailed in Section 4.1 to describe the proposed approach, it is widely applicable to any entity-relation dataset as there are no any data-specific constraints.

3.4. Computational complexity

Finally, we analyse the computational complexity of the SNOWTEC pipeline for information extraction. Following the algorithmic description in Algorithm 1, let n denote the number of tokens in sentence S and $m = |E_{\text{raw}}|$ denote the number of entities detected by the entity classifier. Each classifier for entity and relation extraction consists of L layers and d hidden dimensions. We unfold the complexities associated with each component below.

Step 1: Entity classification. The entity classifier is a transformer encoder with a sequence labelling head (Section 3.2). As shown in Vaswani et al. (2017), the self-attention mechanism incurs $\mathcal{O}(n^2 \cdot d)$ computational complexity per layer, resulting in $\mathcal{O}(n^2 \cdot d \cdot L)$ for the entire transformer encoder. The token classification head with softmax layers adds $\mathcal{O}(n \cdot k_e)$ complexity, where $k_e = |EntityTypes|$, the number of entity categories. Since the complexity is dominated by the transformer encoder, the overall complexity of this step is $\mathcal{O}(n^2 \cdot d \cdot L)$.

Step 2: Entity pairing. The entity pairing step enumerates unordered, unique entity pairs, resulting in a total of $\frac{m(m-1)}{2}$ candidate pairs. This enumeration runs in $\mathcal{O}(m^2)$ complexity.

Step 3: Relation classification. For each candidate entity pair, the relation classifier processes the full sentence S, through a transformer encoder (Section 3.3). This step results $\mathcal{O}(n^2 \cdot d \cdot L)$ complexity, similar to the encoder block in the entity classifier. The entity embeddings are then concatenated and passed through a softmax layer, adding $\mathcal{O}(k_r)$ complexity per pair ($k_r = |RelationTypes|$, the number of relation categories), which is negligible compared to the transformer encoder, and therefore can be asymptotically ignored. As a result, the total complexity of this step is $\mathcal{O}(m^2 \cdot n^2 \cdot d \cdot L)$.

Step 4: Graph construction. The final step assembles the graph from m entity nodes and at most $\frac{m(m-1)}{2}$ relation edges, contributing $\mathcal{O}(m^2)$ complexity.

Overall, the pipeline is dominated by the repeated transformer inference in step 3 for relation classification, returning a complexity of $\mathcal{O}(m^2 \cdot n^2 \cdot d \cdot L)$. In practice, $m \ll n$ for typical regulatory sentences, since the number of detected entities is small relative to the sentence length. Under this assumption, the quadratic factor m^2 remains modest, simplifying the overall cost to $\mathcal{O}(n^2 \cdot d \cdot L)$.

4. Experimental setup

In our experiments, we utilised the CODE-ACCORD dataset (Hettiarachchi et al., 2025) and a regulatory text corpus extracted from England's approved document collection (Section 4.1). Further details regarding the pre-trained transformer/language models employed in conjunction with our proposed architectures can be found in Section 4.2, along with their respective hyper-parameter settings. Section 4.3 outlines the metrics involved in the performance evaluation of models. Our implementations were conducted in Python, leveraging PyTorch (Paszke et al., 2019) and Huggingface (Wolf et al., 2020) libraries to build machine learning models. For the visualisation of graph outputs, we used Graphviz library.[2] All experiments were performed using an Intel Xeon Gold 6240 GPU.

4.1. Data

4.1.1. Annotated data

We use the CODE-ACCORD dataset (Hettiarachchi et al., 2025) to evaluate the proposed transformer-based models. It comprises regulatory sentences extracted from the building regulations of England and Finland across 18 and 8 sub-domains, respectively, including accessibility, fire safety, structural integrity and energy efficiency. All

[2] The documentation of Graphviz is available on https://graphviz.readthedocs.io/en/stable/.

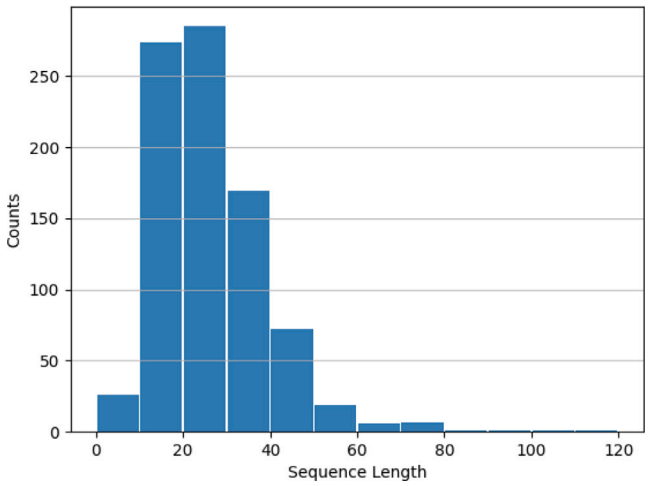


Fig. 6. Sequence length distribution of annotated sentences in the CODE-ACCORD dataset.

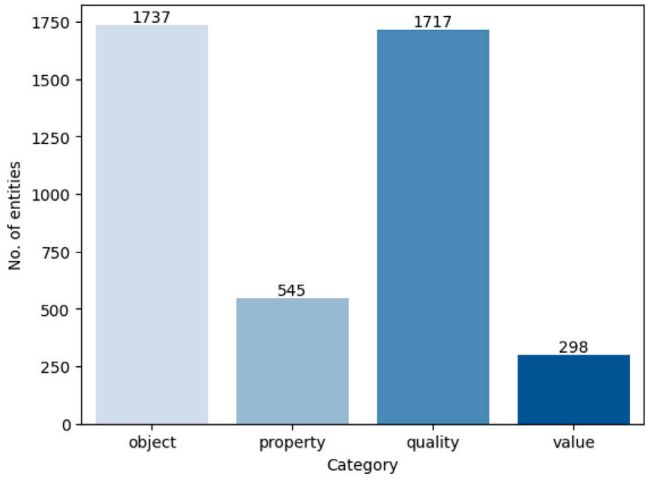


Fig. 7. Distribution of entity categories in the CODE-ACCORD dataset.

the sentences are in English, as the English translation of the Finnish regulations is used.

Each sentence is annotated with entities, representing regulatory components (e.g. window, smoke detectors, etc.) and relations, capturing the semantic associations between the entities. Together, the annotations form a sentence-level knowledge graph that structurally represents the regulatory requirement expressed in natural language. Importantly, the annotation schema is designed to be domain-general rather than tied to a specific regulatory chapter or requirement type, enabling broader applicability across heterogeneous regulatory contexts within the AEC sector.

As of now, to the best of our knowledge, CODE-ACCORD stands as the sole publicly available dataset offering comprehensive entity and relation annotations specifically designed for building regulatory data. Its multi-country and multi-subdomain coverage enables systematic evaluation of model generalisability, distinguishing it from earlier data designs that are restricted to narrow regulatory scopes (Zhang & El-Gohary, 2017; Zhou et al., 2022). More details about the entity and relation categories used in CODE-ACCORD are available in Appendix.

Within CODE-ACCORD, there are 862 annotated regulatory sentences. Fig. 6 illustrates the sequence length distribution of these sentences, revealing that a majority consist of fewer than 40 tokens. The dataset comprises a total of 4297 entity annotations across four categories (Fig. 7). Covering relations, there are 4329 annotations across

Table 2
Statistics of original and augmented relation-annotated training data.

Category	Original data	Synthetic data	Total
Equal	59	708	767
Greater	44	528	572
Greater-equal	86	800	886
Less	7	84	91
Less-equal	56	672	728
Necessity	822	–	822
None	800	–	800
Not-part-of	10	120	130
Part-of	650	–	650
Selection	929	–	929
Training data	3463	2912	6375

10 categories (Fig. 8). For both proposed classifiers, we considered 80% of the samples as training data and the remaining 20% as testing data. However, this results in a different number of samples per model as the entity classifier treats each sentence as an independent sample, while the relation classifier treats individual relations as separate samples.

Data augmentation:. Considering the high imbalance nature within relation distribution, primarily characterised by underrepresented categories, we applied the augmentation technique outlined in Section 3.3.1 for relation-annotated data. We only focused on generating synthetic data for categories: 'equal', 'greater', 'greater-equal', 'less', 'less-equal' and 'not-part-of', which exhibited a scarcity in representation. Importantly, the test set remained unaltered while we exclusively augmented the training data. For each sample in the training set, we generated 12 synthetic samples, considering the overall categorical distribution. Altogether, data augmentation increased the original training dataset with 3463 samples by 2912 samples, as summarised in Table 2. Moreover, Fig. 9(b) illustrates the distribution of final training dataset, following the original distribution in Fig. 9(a).

4.1.2. Regulatory text corpus

For language modelling, we developed a text corpus using England's approved document collection which provides guidance to meet building regulations.[3] Initially, we transformed the PDF files into text format using the PDFMiner library.[4] Subsequently, we conducted a series of data-cleaning procedures. This involved the rectification of formatting inconsistencies that occurred during the conversion process and the elimination of non-semantic elements such as pointers (e.g., G1, 1.2, a., etc.). Additionally, we filtered out section headings, incomplete text segments and those containing fewer than three words, exclusively retaining complete sentences and bullet-pointed text for the refined corpus. The final dataset comprised 14,336 text segments totalling 302,469 tokens. Fig. 10 illustrates the sequence length distribution within this refined corpus. We split the data using 80%–20% split to have training and testing data.

4.2. Transformers and hyper-parameters

We involved four transformer models: (1) BERT-LARGE (bert-large-cased) (Devlin et al., 2019), (2) ROBERTA-LARGE (roberta-large) (Liu et al., 2019), (3) ALBERT-LARGE (albert-large-v2) (Lan et al., 2020), and (4) ELECTRA-LARGE (electra-large-discriminator) (Clark et al., 2020), pre-trained on English text for our experiments. These models have gained significant popularity in recent research due to their outstanding performance (Han et al., 2021; Hettiarachchi & Ranasinghe, 2023). Also, these models were pre-trained in different settings. ROBERTA is a BERT

[3] England's approved document collection is available on https://www.gov.uk/government/collections/approved-documents.

[4] PDFMiner documentation is available on https://pypi.org/project/pdfminer/.

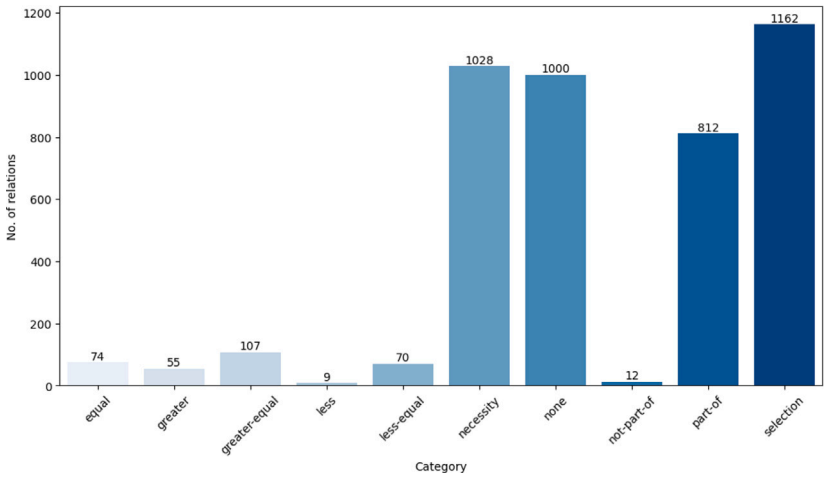
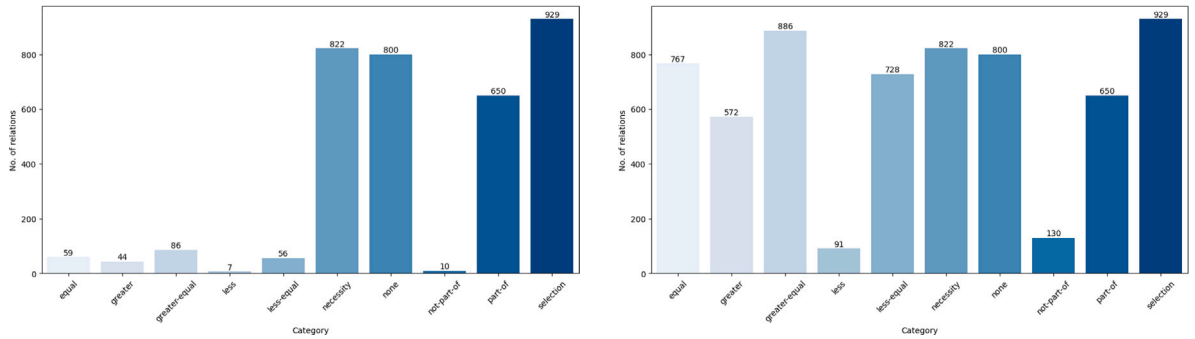


Fig. 8. Distribution of relation categories in the CODE-ACCORD dataset.



(a) Original data (b) Augmented data

Fig. 9. Distribution of the relation categories in original and augmented relation-annotated training data.

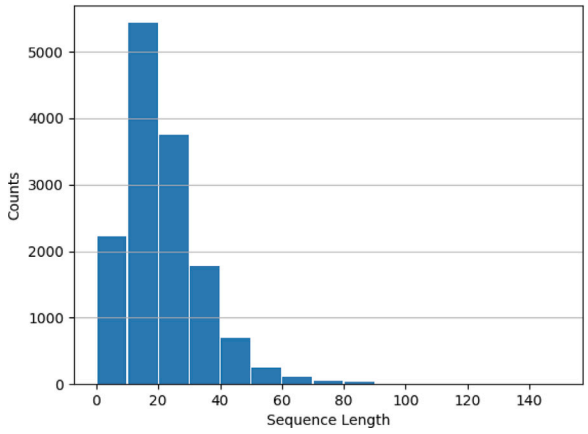


Fig. 10. Sequence length distribution of text elements in the raw text corpus.

variant, which is trained on more data with longer training sentences. ALBERT is a lite version of BERT with parameter reduction to speed up the training process. ELECTRA follows a token discrimination-based pre-training process to facilitate more efficient training. All these pre-trained transformers were obtained from HuggingFace's model repository (Wolf et al., 2020) to conduct the experiments.

Our experiments used a common hyper-parameter setup to generate comparable results while maintaining consistency among different architectures. Also, we believe these parameters will provide a basis for future experiments. For all entity and relation classifier models, we set the maximum sequence length to 128, following the sequence length distribution of targeted text data (Fig. 6). The batch size of 16, the learning rate of $1e^{-5}$ with Adam optimiser and five epochs with early stopping patience of 10 were used considering the computational complexities of transformers. Depending on the training data availability, we set the evaluation steps to eight for the entity classifier and 16 for the relation classifier, allowing five and 15 evaluations per epoch, respectively. These evaluations were conducted on a validation split, which is 10% from training data, while using the remaining 90% for training.

We applied two model optimisation techniques: (1) hyper-parameter tuning and (2) language modelling, during our experiments. During hyper-parameter tuning, we particularly focused on optimising the model's learning rate. Our initial experiments revealed that adjusting this parameter yielded a more noticeable impact on the final results compared to others. More details about our approach are discussed in Section 5 together with each classifier's results. For language modelling, we used the maximum sequence length of 128 following the distribution in Fig. 10. We set the batch size to 16 and the learning rate to $3e^{-5}$ with Adam optimiser, following our initial experiments. We allowed the model to train on 25 epochs with an early stopping patience of 10, considering the sensitivity of this task.

4.3. Evaluation metrics

To evaluate the model's performance, we used precision (P), recall (R) and F1 scores following Eqs. (2)–(4). In these equations, TP, FP and FN refer to the true positive, false positive and false negative counts,

respectively. Precision measures the accuracy of the predictions made by the model, while recall measures the portion of actual instances that were correctly predicted by the model. F1 is the weighted harmonic mean of precision and recall, combining their properties.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

For entity evaluation, we employed two approaches: (1) strict span-and-type matching and (2) token-level matching. For strict evaluation, we used the Seqeval framework with strict mode on BIO format Nakayama (2018). This approach compares the text span, entity type, and BIO label to compute TP, FP and FN values, and considers a span correct (TP) only if all elements exactly match the ground truth. While this method captures the ideal scenario, it can be overly strict for multi-token entities, as even a single token offset can result in the entire prediction being marked incorrect, potentially exaggerating semantic failures. To address this, we also incorporated token-level matching, which is more flexible. In this setup, each token is evaluated independently, and a prediction is considered correct (TP) if the predicted entity category matches the true category, regardless of BIO tagging or full-span alignment.

For relation evaluation, TP, FP, and FN values were computed solely by comparing predicted and actual categories. A prediction is marked correct (TP) only if it perfectly matches the ground truth category.

Given the equal importance of multiple categories in both entity and relation classification, we employed the macro-averaging method to derive the final metrics. This involved initially computing P, R and F1 values individually for each category. Subsequently, we computed their unweighted mean to obtain the final metrics, following Equation (5). In this equation, X represents a metric (i.e. P, R or F1), n represents the total number of categories, and X_i represents a per-class metric value.

$$Macro\ X = \frac{\sum_{i=1}^{n} X_i}{n} \tag{5}$$

5. Results and discussion

We present the performance evaluations of the machine learning models developed in this research: entity classifier and relation classifier, in Sections 5.1 and 5.2. These assessments include in-depth error analyses aimed at understanding models' behaviour. Subsequently, in Section 5.3, we offer a demonstration of the complete functionality of the proposed SNOWTEC pipeline. This demonstration utilises real-world regulatory sentences, showcasing the practical application and effectiveness of our approach.

5.1. Entity classifier

For entity classification, we evaluated the performance of the proposed architecture (Fig. 4) using four pre-trained transformers described in Section 4.2. The results obtained on validation and test datasets are summarised in Table 3. As the model is fine-tuned based on the results of validation data, which comprises a smaller subset (10% of the training data) compared to the test data, the validation scores are expected to be slightly higher than those on the test set. This trend is observed across most models. To provide a more comprehensive evaluation, we additionally report token-level metrics on the test data. As expected, token-level evaluation yields higher scores than strict (entity-level) evaluation due to its less restrictive matching criteria. This discrepancy suggests that, although entity boundaries of predictions may occasionally deviate from the gold standard, the models are generally effective at identifying the correct entity tokens.

Among the transformer models, ROBERTA-LARGE achieves the best overall performance. It outperforms the other models by an average margin of 4.2% in validation F1 score, 8.3% in strict-mode test F1 score, and 7.2% in token-level test F1 score. These results also suggest that if the language model is pre-trained on a large corpus, it can be fine-tuned for a downstream task effectively.

Although the ROBERTA-LARGE model achieves the best F1 score along with recall, it does not consistently achieve the highest precision. All the other models exhibit higher precision than recall, occasionally even surpassing the precision of ROBERTA-LARGE. This performance asymmetry can be directly traceable to the distinct pre-training objectives of each transformer model. BERT, ELECTRA, and ALBERT were pretrained with secondary objectives (e.g., inter-sentence or token anomaly detection objectives) alongside masked language modelling (Clark et al., 2020; Devlin et al., 2019; Lan et al., 2020). The combined objectives tend to make these models more conservative, leading to stricter entity prediction, resulting in high precision with comparatively low recall. In contrast, ROBERTA model is pre-trained solely using dynamic masking (Liu et al., 2019). This design optimises the model entirely on contextual token prediction, potentially enabling broader contextual generalisation. When fine-tuned for entity recognition, as suggested by the results, this design appears to allow the model to identify a wider range of true positives at the cost of occasionally over-predicting entities, thereby reducing precision. In the context of regulatory information extraction, this trade-off is operationally preferable, as missing true instances has a greater downstream cost than predicting false positives, which can be filtered at later stages.

We further optimised our best-performing entity classifier, and our findings are summarised in Table 4. As described in Section 4.2, within hyper-parameter tuning, our primary focus was centred on optimising the learning rate. We experimented with additional rates ($1e^{-3}$, $1e^{-4}$, and $1e^{-6}$) following the initial setup of $1e^{-5}$. The corresponding training and validation curves are visualised in Fig. 11. Notably, a learning rate of $1e^{-4}$ performed on par with $1e^{-5}$ but showcased better convergence. Fine-tuning the learning rate resulted in substantial performance gains, a 9.1% increase in validation F1 score and an 8.2% improvement in strict-mode test F1 score. Additionally, the token-level test F1 score has improved by 2.6%.

Furthermore, we applied language modelling to the ROBERTA-LARGE model using the regulatory text corpus described in Section 4.1.2. As shown in Table 4, this step contributed to an additional improvement of 0.6% in validation F1 score (strict mode) and a 1.5% increase in token-level test F1 score. However, this resulted in a slight 0.9% drop in the strict-mode test F1 score. Overall, these modest and partially inconsistent gains suggest that the size of the domain-specific corpus may limit the effectiveness of continued pre-training. Leveraging a larger regulatory corpus could lead to more substantial enhancements, aligning with other recent studies (Gururangan et al., 2020; Ranasinghe et al., 2025).

When comparing optimisation strategies applied to ROBERTA-LARGE, language modelling achieves the best overall F1 score, although its precision on both validation and test sets is slightly lower than that obtained with hyper-parameter tuning. Hyper-parameter tuning focuses on optimising decision boundaries to produce a more controlled, accurate prediction distribution. In contrast, language modelling on in-domain text (domain-adaptive pre-training) encourages the model to learn domain-specific lexical and syntactic patterns, increasing its likelihood of recognising entities it would otherwise miss in a general setting. This introduces the same precision–recall tradeoff, which we observed in the general ROBERTA-LARGE model compared to other pre-trained transformers. As the model becomes more entity-aware, it appears to be more prone to borderline cases, modestly reducing precision while increasing recall. Nonetheless, our findings indicate that imparting domain knowledge to language models helps in enhancing entity classification performance. We further discuss the performance of the best entity classifier in Section 5.1.1, providing an in-depth analysis.

Table 3
Performance evaluation of the entity classifier across validation and test datasets using various transformer models. The best F1 score is marked in bold.

Transformer	Validation (Strict)			Test (Strict)			Test (Token-level)		
	P	R	F1	P	R	F1	P	R	F1
BERT-LARGE	0.6182	0.5448	0.5686	0.6288	0.5446	0.5694	0.8277	0.7046	0.7514
ELECTRA-LARGE	0.6838	0.5486	0.5964	0.5633	0.4468	0.4740	0.7312	0.5747	0.6157
ALBERT-LARGE	0.6708	0.5841	0.6219	0.6270	0.5354	0.5703	0.8088	0.7174	0.7547
ROBERTA-LARGE	0.6352	0.6427	**0.6379**	0.6281	0.6197	**0.6212**	0.8090	0.7593	**0.7790**

Table 4
Impact on best-performed entity classifier's (ROBERTA-LARGE) results by different optimisation techniques. The best F1 score is marked in bold.

Optimisation Technique	Validation (Strict)			Test (Strict)			Test (Token-level)		
	P	R	F1	P	R	F1	P	R	F1
–	0.6352	0.6427	0.6379	0.6281	0.6197	0.6212	0.8090	0.7593	0.7790
Hyper-parameter tuning	0.7674	0.6956	0.7287	0.7708	0.6550	**0.7028**	0.8689	0.7642	0.8054
Language modelling	0.7655	0.7100	**0.7351**	0.7244	0.6678	0.6940	0.8607	0.7884	**0.8207**



(a) Training loss (b) Evaluation loss

Fig. 11. Training and validation learning curves of entity classifiers built using ROBERTA-LARGE model on different learning rates.

5.1.1. Error analysis

We further analysed the best-performed entity classifier (i.e. ROBERTA-LARGE with language modelling) results to identify the causes of its performance limitations. Our analysis revealed two major factors: (1) unique attributes of entity categories and (2) contextual complexities, that directly affect the model performance, as elaborated below.

Impact by unique attributes of entity categories:. In entity classification, we aim to identify four categories (i.e. object, property, quality and value) defined in the CODE-ACCORD dataset (Section 4.1.1), which hold their unique characteristics. For instance, objects and values were mostly formed by one or two tokens/words. Additionally, a high proportion of values have numbers mostly coupled with units. In contrast, properties and qualities have a composition of short and long text sequences. Overall, qualities have more lengthy text spans than all other categories. We showcase a few randomly selected entities from each category in Table 5 to further emphasise their distinctive attributes.

Table 6 presents the outcomes of our analysis regarding the model's predictive performance for each entity category, evaluated under both strict and token-level matching criteria. Strict matching requires an exact span boundary match, while token-level matching awards partial credit for overlapping predictions, offering a more flexible assessment of near-correct extractions. Under strict matching, objects and values, the entity categories characterised by more definitive structures with shorter sequences, demonstrated comparatively higher F1 scores (0.7136 and 0.8261, respectively, on the test set). Since values have the fewest annotated samples (Fig. 7), these results suggest the integrity of the entity structure has more influence on the model's learning than the sample count. This fact is further confirmed by the results obtained in quality. Even though quality has more annotated samples (×6 than value samples), it resulted in the lowest F1 score (0.5732 on the test set) due to its complex structure.

The token-level results reveal an additional and informative pattern. Quality shows the largest improvement between evaluation modes, rising from 0.5732 under strict matching to 0.8735 under token-level matching on the test set. This suggests that while the model frequently captures the correct tokens belonging to quality entities, it struggles with precise boundary detection, which is a natural consequence of quality spans being longer and more structurally variable. In contrast, value and object categories show comparatively smaller gains between the two modes, consistent with their shorter, more bounded nature, where strict and token-level assessments tend to converge.

Overall, our findings suggest that the inherent nature or structure of an entity category significantly influences the learning process of a transformer-based entity classifier, often surpassing the impact of the sample count within the training data. The model requires more data from complex entity categories to learn them better, while a smaller amount of data is sufficient to learn the simple (or more structured) entities.

Impact by the contextual complexities:. In natural language, the organisational structure of textual elements within a context significantly influences entity identification. To measure contextual complexity in our experiments, we used the entity count within a sentence, where

Table 5
Entity categories with a few samples.

Object	Property	Quality	Value
Building	Height	Mechanical	One
Doors	Width	Insulated	900 mm
Parking spaces	U-values	Wheelchair-accessible	500 millimetres
Windows	Target primary energy rate	For entering a dwelling	0.7 metres
Ventilation systems	Floor area	Insulated at ceiling level	0.15 m/s

Table 6
Results of the best-performed entity classifier (ROBERTA-LARGE) on validation and test datasets across different entity categories. Strict matching requires exact span boundary agreement, whereas token-level matching assigns partial credit for overlapping tokens, leading to higher support counts as tokens (not complete entity spans) are evaluated.

Category	Validation				Test			
	Support	P	R	F1	Support	P	R	F1
	Strict Matching							
Object	146	0.7955	0.7192	0.7554	317	0.7240	0.7035	0.7136
Property	48	0.6154	0.6667	0.6400	98	0.7375	0.6020	0.6629
Quality	139	0.7281	0.5971	0.6561	317	0.5723	0.5741	0.5732
Value	28	0.9231	0.8571	0.8889	48	0.8636	0.7917	0.8261
Macro average	361	0.7655	0.7100	0.7351	780	0.7244	0.6678	0.6940
	Token-level Matching							
Object	254	0.9104	0.7598	0.8283	506	0.8351	0.7905	0.8122
Property	100	0.7596	0.7900	0.7745	189	0.8114	0.7513	0.7802
Quality	607	0.8292	0.9275	0.8756	1923	0.8607	0.8866	0.8735
Value	56	0.9804	0.8929	0.9346	80	0.9355	0.7250	0.8169
Macro average	1017	0.8699	0.8426	0.8532	2698	0.8607	0.7884	0.8207

a higher count denotes a more detailed and intricate context. Table 7 summarises the entity classifier's performance variations across complexity levels, evaluated under both strict and token-level matching criteria.

Under strict matching, the results indicate a more complex pattern that cannot be explained by a straightforward complexity–performance trade-off. Sentences with higher entity counts (> 5) obtain the highest macro F1 (0.7354), followed by the moderate-complexity group (sentences with 3–5 entities), with the low-complexity group (sentences with entities ≤ 3) achieving the lowest macro F1 (0.6361). This suggests that while very simple sentences may lack sufficient contextual signal for precise boundary detection, moderately and highly complex sentences provide richer context that the model can exploit. Token-level matching scores are consistently high across all complexity groups (macro F1 ranging from 0.7855 to 0.8535), indicating that the model reliably identifies the correct tokens regardless of sentence complexity, with boundary detection being the primary source of strict matching errors.

Overall, these findings suggest that contextual richness, rather than simplicity alone, can support entity classification. The model may benefit from additional exposure to complex sentence structures to improve boundary detection of entity categories.

5.2. Relation classifier

For relation classification, we evaluated the performance of the proposed architecture (Fig. 5) using three pre-trained transformers described in Section 4.2. Considering the limited performance observed during entity classification experiments, the ELECTRA-LARGE model was excluded from relation classification. The summary of results obtained from both validation and test datasets is presented in Table 8. Similar to the trends observed in the entity classifier results (Table 3), the validation set reflects more accurate predictions compared to the test set. The standard practices of model fine-tuning based on validation outcomes and the smaller size of the validation set compared to the test set mainly result in such trends.

Overall, Table 8 demonstrates a significant enhancement in the performance of the relation classifier achieved through data augmentation. On average, all models experienced a noteworthy 26% increase in F1 scores after integrating augmented data. Similar to observations in the entity classifier scenario, among the transformer models, ROBERTA-LARGE exhibited superior performance. Initially, using the original dataset, ROBERTA-LARGE showcased an average increase of 5.4% in validation F1 scores and 3.4% in test F1 scores compared to other models. Even after incorporating augmented data, ROBERTA-LARGE maintained its lead, exhibiting an average increase of 2.9% in validation F1 scores and 2.3% in test F1 scores. These outcomes reinforce the notion that pre-training a language model on a substantial corpus enables effective fine-tuning for downstream tasks, consistent with our observations in entity classifiers. Additionally, the performance of the BERT-LARGE model approached that of ROBERTA-LARGE following data augmentation. This highlights that an increased volume of data for a downstream task (i.e. relation classification) facilitates the model in learning task-specific details, minimising reliance on the original knowledge of the language model.

Moreover, the precision–recall asymmetry observed across the transformer models in relation classification is consistent with the pattern observed in the entity classifiers' results. As discussed in Section 5.1, the ROBERTA model's sole reliance on masked language modelling, without an additional pre-training objective, produces richer token-level representations. In relation classification, this influences how the model encodes entity pairs and their surrounding context, enabling it to recognise a broader range of relations, resulting in higher recall at the cost of slightly lower precision than other models.

We further optimised our best-performing relation classifier using the same techniques we used with the entity classifiers (i.e. hyper-parameter tuning and language modelling). Table 9 provides an overview of the obtained results. Under hyper-parameter tuning, we explored the model's performance by experimenting with three additional learning rates ($1e^{-3}$, $1e^{-4}$, and $1e^{-6}$) subsequent to the initial setup of $1e^{-5}$. However, upon reviewing the corresponding training and validation curves depicted in Fig. 12, $1e^{-5}$ turned out as the optimal learning rate.

In language modelling, we employed the same ROBERTA-LARGE model trained on the regulatory text corpus detailed in Section 4.1.2, utilised

Table 7
Results of the best-performed entity classifier (ROBERTA-LARGE) on test datasets depending on the entity count per sentence.

Entity count	Category	Strict matching				Token-level matching			
		Support	P	R	F1	Support	P	R	F1
≤3	Object	70	0.6667	0.7429	0.7027	125	0.8095	0.8160	0.8127
	Property	17	0.7143	0.5882	0.6452	35	0.7000	0.8000	0.7467
	Quality	87	0.5652	0.5977	0.5810	721	0.943	0.9071	0.9250
	Value	7	0.6667	0.5714	0.6154	12	0.8182	0.7500	0.7826
	Macro average	181	0.6532	0.6251	0.6361	893	0.8179	0.8183	0.8168
>3 and ≤5	Object	100	0.7100	0.7100	0.7100	155	0.8101	0.8258	0.8179
	Property	44	0.6944	0.5682	0.6250	72	0.7727	0.7083	0.7391
	Quality	103	0.5556	0.6311	0.5909	651	0.8375	0.8786	0.8576
	Value	19	0.7778	0.7368	0.7568	40	0.9231	0.6000	0.7273
	Macro average	266	0.6844	0.6615	0.6707	918	0.8359	0.7532	0.7855
>5	Object	147	0.7692	0.6803	0.7220	226	0.8718	0.7522	0.8076
	Property	37	0.8000	0.6486	0.7164	82	0.9130	0.7683	0.8344
	Quality	127	0.5963	0.5118	0.5508	551	0.7917	0.8693	0.8287
	Value	22	1.0000	0.9091	0.9524	28	1.0000	0.8929	0.9434
	Macro average	333	0.7914	0.6875	0.7354	887	0.8941	0.8207	0.8535

Table 8
Performance evaluation of the relation classifier across validation and test datasets using various transformer models, with and without the integration of data augmentation. The best F1 score is marked in bold. The improvement in F1 score on the test data after applying data augmentation is indicated within brackets.

Transformer	Validation			Test		
	P	R	F1	P	R	F1
BERT-LARGE	0.6129	0.4887	0.5144	0.5714	0.5029	0.5243
ALBERT-LARGE	0.6332	0.5393	0.5590	0.5592	0.4978	0.5122
ROBERTA-LARGE	0.6099	0.5895	**0.5903**	0.5577	0.5481	**0.5498**
Data Augmentation						
BERT-LARGE	0.9215	0.9239	0.9207	0.8280	0.8161	0.8009 (+28%)
ALBERT-LARGE	0.9098	0.9128	0.9109	0.8060	0.7511	0.7556 (+24%)
ROBERTA-LARGE	0.9448	0.9459	**0.9450**	0.7997	0.8352	**0.8011 (+25%)**

Table 9
Impact on best-performed relation classifier's (ROBERTA-LARGE) results by different optimisation techniques. The best F1 score is marked in bold.

Optimisation technique	Validation			Test		
	P	R	F1	P	R	F1
–	0.9448	0.9459	**0.9450**	0.7997	0.8352	**0.8011**
Hyper-parameter tuning	0.9448	0.9459	**0.9450**	0.7997	0.8352	**0.8011**
Language modelling	0.9441	0.9461	0.9447	0.8507	0.7722	0.7922

for the entity classification experiments. Surprisingly, unlike the entity classification results, the relation classification outcomes exhibited a slight decrease following language modelling. This suggests a potential risk of confusion when applying language modelling using a limited domain-specific corpus alongside a rich training dataset for the downstream task. Overall, the application of optimisation techniques did not improve the relation classifier's performance. Nonetheless, we conducted an in-depth analysis of the best relation classifier, and our findings are discussed in Section 5.2.1.

5.2.1. Error analysis

We further analysed the best-performed relation classifier (i.e. ROBERTA-LARGE with data augmentation) results to identify the factors that hinder its performance. The overall assessment displayed consistent performance across all relation categories, with a validation F1 score exceeding 80%, as depicted in Table 10. Although there were slight declines in the greater-equal and less-equal categories in the test set, given their association with a relatively small sample count, these drops can be deemed negligible. Altogether, these findings suggest that there is no notable impact from the relation category on the model's final performance.

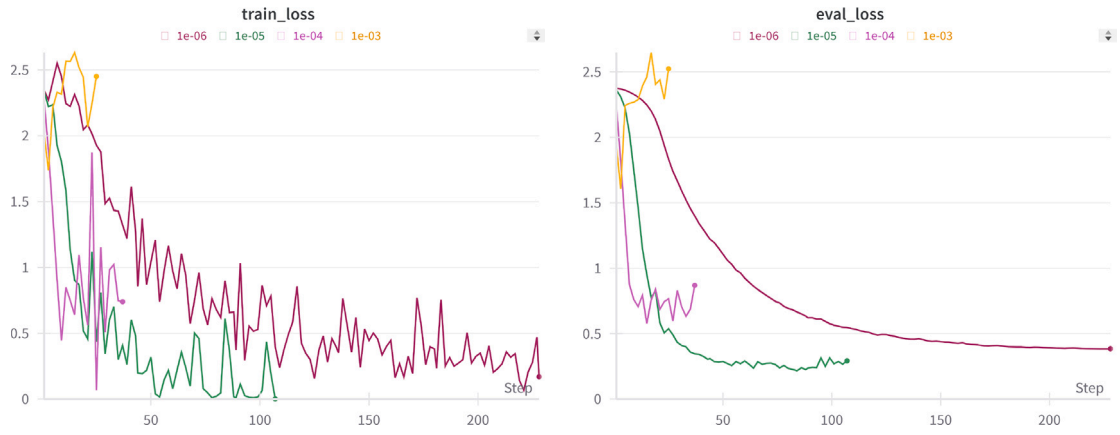
Secondly, we explored the misclassifications to identify potential confusion between relation categories. This in-depth analysis revealed that complex textual structures influence the model's performance, leading to misclassification. Details of these findings are outlined below.

Impact by complex textual structures:. The confusion matrix of the best-performed relation classifier on the test dataset is illustrated in Fig. 13. This matrix exhibits clear patterns, confirming the model's overall strong performance. Diagonal dominance signifies a majority of accurate predictions, accompanied by low off-diagonal values, which indicate minimal misclassification. However, we further investigated the misclassifications (particularly, the off-diagonal values above 10) to understand the factors contributing to the model's confusion. Our analysis pinpointed that most of these instances of confusion arose from complexities within the textual structure. To explain this finding, we summarise the frequently identified various textual complexities in Table 11 together with relevant data samples.

5.3. SNOWTEC demonstration

Finally, we demonstrate the complete functionality of the SNOWTEC pipeline (Fig. 1) leveraging the best-performing entity and relation classifiers detailed in Sections 5.1 and 5.2, respectively. SNOWTEC automatically identifies entities and their relations within regulatory sentences to generate knowledge graphs that encapsulate the conveyed information.

Table 12 presents a selection of tested samples processed through the finalised pipeline, each accompanied by its generated graph. As can be seen, these graphs comprehensively capture the ideas expressed in the original sentences in a structured manner. Also, these graphs serve as a machine-processable output, capturing and processing the linguistic complexities inherent in regulatory sentences. The accuracy



(a) Training loss (b) Evaluation loss

Fig. 12. Training and validation learning curves of relation classifiers built using ROBERTA-LARGE model on different learning rates.

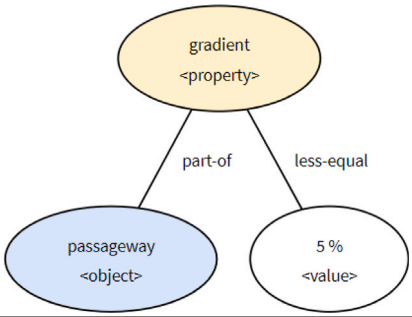
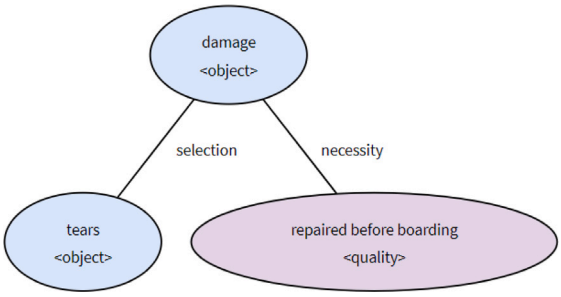
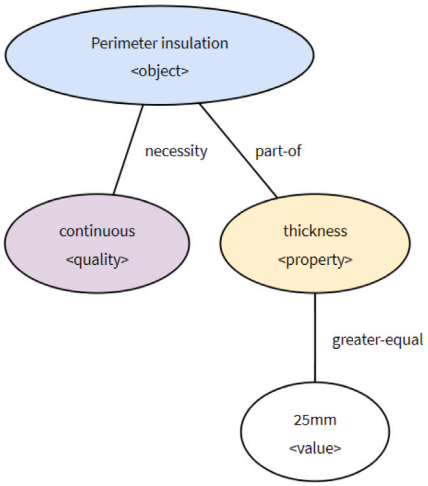
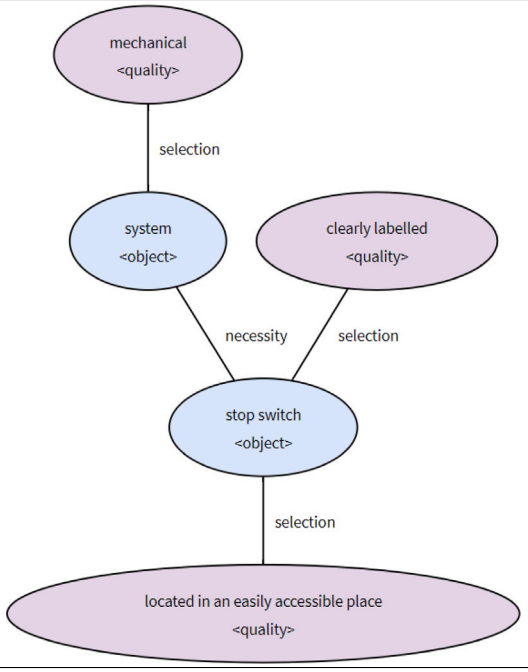
Table 10
Results of the best-performed relation classifier (ROBERTA-LARGE) on validation and test datasets across different relation categories.

Category	Validation				Test			
	Support	P	R	F1	Support	P	R	F1
Equal	77	0.9625	1.0000	0.9809	15	0.7143	1.0000	0.8333
Greater	57	1.0000	1.0000	1.0000	11	0.8182	0.8182	0.8182
Greater-equal	89	0.9889	1.0000	0.9944	21	0.7273	0.7619	0.7442
Less	9	1.0000	1.0000	1.0000	2	1.0000	1.0000	1.0000
Less-equal	73	0.9865	1.0000	0.9932	14	0.6842	0.9286	0.7879
Necessity	82	0.8902	0.8902	0.8902	206	0.9167	0.8544	0.8844
None	80	0.8592	0.7625	0.8079	200	0.8289	0.7750	0.8010
Not-part-of	13	1.0000	1.0000	1.0000	2	0.6667	1.0000	0.8000
Part-of	65	0.8657	0.8923	0.8788	162	0.7910	0.8642	0.8260
Selection	93	0.8947	0.9140	0.9043	233	0.8498	0.8498	0.8498
Macro average	638	0.9448	0.9459	0.9450	866	0.7997	0.8352	0.8011

Table 11
Frequently misclassified relations with identified causes and samples.

True label	Predicted label	Sample	Reason
Necessity	None	Any ⟨e1⟩walls⟨/e1⟩, doors and windows should be insulated and ⟨e2⟩draught-proofed⟨/e2⟩ to at least the same extent as in the existing building. Ground arrays including ⟨e1⟩header pipes⟨/e1⟩ and manifolds should be ⟨e2⟩flushed as one system to remove all debris and purged to remove all air⟨/e2⟩.	Multiple necessities
None	Part-of	The ⟨e1⟩roof structures⟨/e1⟩ and ⟨e2⟩joints⟨/e2⟩ must be properly inclined and sealed for water drainage. The natural or mechanical ⟨e1⟩ventilation system⟨/e1⟩ of a building shall be strong and its ⟨e2⟩air-tightness⟨/e2⟩ shall be at least class B.	Conjoined elements coreferences
None	Selection	There shall be ⟨e1⟩adequate means⟨/e1⟩ of ventilation ⟨e2⟩provided for people in the building⟨/e2⟩. The air extract rate should be 20 litres per second per ⟨e1⟩machine⟨/e1⟩ ⟨e2⟩during use⟨/e2⟩.	Hierarchical relationships
Selection	None	In an existing building, for the principal or ⟨e1⟩main staff⟨/e1⟩ entrance or ⟨e2⟩entrances⟨/e2⟩ to be accessible, an alternative accessible entrance should be provided. A building must be provided with a fire wall if ⟨e1⟩it⟨/e1⟩ is situated adjacent to a neighbouring building or ⟨e2⟩so close to a neighbouring building that the spread of fire is evident⟨/e2⟩.	Multiple selections
Selection	Part-of	⟨e1⟩Requirements⟨/e1⟩ for ⟨e2⟩accessibility⟨/e2⟩ should be balanced against preserving historic buildings or environments. The ⟨e1⟩impact⟨/e1⟩ of the ⟨e2⟩ground⟨/e2⟩ and crawl space on heat loss shall be taken into account in heat loss calculations.	Confusing text patterns

Table 12
Knowledge graphs generated by SNOWTEC for a set of sample sentences. In. and Out. denotes the input to and output from the pipeline.

In.	The gradient of the passageway should not exceed 5%.	Any damage, such as tears, should be repaired before boarding.
Out.		
In.	Perimeter insulation should be continuous and have a minimum thickness of 25mm.	In a mechanical system, there shall be a clearly labelled stop switch, which shall be located in an easily accessible place.
Out.		

limitations exhibited in current machine learning models might negatively impact the conversion of sentences with complex structures, requiring models that are enhanced in light of the findings of this work. However, we believe this framework is a strong starting point for converting regulatory text into machine-processable formats utilising the recent NLP techniques. Also, the descriptive nature of entity-relation graphs supports human-in-the-loop validation, which helps mitigate possible errors and encourages more reliable and transparent conversions. Ultimately, SNOWTEC will facilitate effective ACC by converting complex regulatory data into a more simple format.

6. Conclusions and future work

This paper proposed *SNOWTEC*, an innovative information extraction pipeline powered by a synthetic natural language oversampling approach designed to extract regulatory information from text for ACC within the AEC sector. Given a regulatory sentence, it generates a machine-readable output, a knowledge graph detailing entities and their relationships extracted from natural language capturing the underlying idea(s), which can be readily processed to automate compliance checks. Aligning with recent trends in NLP, SNOWTEC employs two machine learning models based on transformer/language model architectures to extract entities and relations from text. Our experimentation involved diverse pre-trained transformers, involving different learning techniques such as fine-tuning and language modelling to comprehensively evaluate the models' performance in varied settings. We also introduced a data augmentation method for relation-annotated data, leveraging domain-specific entities to address data scarcity concerns that often limit the prediction capabilities of language models.

We used the CODE-ACCORD dataset, which contains data spanning across multiple sub-domains within AEC, including accessibility, fire safety, structural integrity, energy efficiency and more, to evaluate the proposed models' performance, covering their general applicability, which is crucial for real-world applications. Among the four pre-trained transformer models tested (i.e. BERT, RoBERTa, ALBERT and ELECTRA), the RoBERTa model showcased superior performance in both entity and relation extraction. Considering the complexities associated with the targeted tasks due to the unstructured nature of natural language, compounded by structural complexities, ambiguities and domain-specific characteristics in the regulatory text, which varies across sub-domains (e.g. structural integrity, fire safety, etc.), we believe the developed models achieved promising results. Comparatively, the entity classifier exhibited a lower performance than the relation classifier's performance. A detailed analysis unveiled that the complexities within entities, due to their unique attributes and contextual influences, mainly contributed to the entity extraction's limited performance. Additionally, limited training data impacted the model's

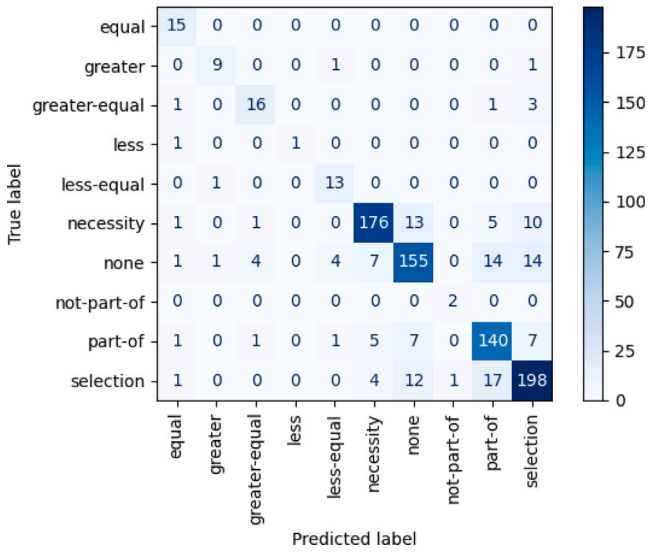


Fig. 13. Confusion matrix of the best-performed relation classifier (ROBERTA-LARGE) on test dataset.

accuracy, notably evidenced by the relation classifier, where data augmentation significantly improved performance by an average increase of 26% in F1. Employing a language modelling approach, enriching the pre-trained language model with domain-specific knowledge from textual data, enhanced the entity classifier's results, being beneficial in scenarios with scarce annotated training data. However, this approach did not yield improvements for the relation classifier, which already had a rich training corpus following the data augmentation. This observed behaviour highlights potential risks of model confusion when employing language modelling using a limited domain-specific corpus alongside a rich training dataset for the downstream task, which will be helpful for future studies in this area.

Overall, we believe our work represents a significant step forward in IE for ACC, as it is solely based on state-of-the-art transformers/language models without requiring any domain-dependent rules. Also, our approach reformulates the IE task as a knowledge graph generation task, further establishing the generalisability across diverse domains. Moreover, knowledge graphs represent information in formats readable by both humans and machines. This enhances the human-in-the-loop reviewing processes, ensuring the accuracy and trustworthiness of AI-generated outputs, which are crucial to ACC. Following our path, in future, we aim to explore the potential of large language models such as Generative Pre-Trained Transformers (GPT) (OpenAI, 2023) and Large Language Model Meta AI (LLaMA) (Touvron et al., 2023) to extract information from regulatory text more effectively. Also, we plan to expand our research to support IE in multiple languages as regulatory documents are written in different languages and translating them into English typically requires substantial manual effort and domain expertise. Additionally, we seek to capture the information expressed by the clausal structure of regulatory documents and interconnections between regulatory sentences, which is crucial for comprehensive rule generation. Furthermore, we hope to align the extracted information with the concepts in a building compliance ontology such as Architecture Engineering and Construction Compliance Checking and Permitting Ontology (AEC3PO)[5] and develop a predictive approach to transform the information into rule languages such as Semantic Web Rule Language (SWRL)[6] and SHApes Constraint Language (SHACL)[7] to facilitate the complete flow of ACC, providing a comprehensive framework.

Table A.13
Entity categories of the CODE-ACCORD dataset.

Entity	Description
Object	An ontological concept which represents a thing that is subject to a particular requirement (e.g. window, fire door)
Property	Property of an object (e.g., width, height)
Quality	Quality or uncountable characteristic of an object/property (e.g. horizontal, self-closing)
Value	A standard or a numerical value that defines a quantity (e.g. 1500 millimetres, five per cent)

CRediT authorship contribution statement

Hansi Hettiarachchi: Conceptualisation, Investigation, Methodology, Software, Validation, Visualisation, Writing – original draft. **Mohamed Medhat Gaber:** Conceptualisation, Methodology, Supervision, Writing – review & editing. **Pouyan Parsafard:** Methodology, Software, Validation. **Edlira Vakaj:** Conceptualisation, Funding acquisition, Project administration, Writing – review & editing.

Declaration of competing 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.

Acknowledgements

This work is funded by the European Union's Horizon Europe research and innovation programme under grant agreement no. 101056973 (ACCORD). UK Participants in the Horizon Europe Project [ACCORD] are supported by UKRI grant numbers [10040207] (Cardiff University), [10038999] (Birmingham City University) and [10049977] (Building Smart International). The computational experiments in this paper were conducted on the computing cluster managed by the Faculty of Computing, Engineering and the Built Environment of Birmingham City University.

Appendix. Entity and relation categories in CODE-ACCORD dataset

Entity and relation categories available in the CODE-ACCORD dataset (Hettiarachchi et al., 2025) are summarised in Tables A.13 and A.14, respectively.

Data availability

The datasets used in this study are publicly available on https://github.com/Accord-Project/CODE-ACCORD/.

[5] AEC3PO ontology documentation is available on https://accordproject.eu/building-compliance-ontology-released/.

[6] SWRL documentation is available on http://www.w3.org/Submission/SWRL/.

[7] SHACL documentation is available on https://www.w3.org/TR/shacl/.

Table A.14
Relation categories of the CODE-ACCORD dataset.

Relation	Description
Selection	A limit to the scope of an object/property based on another object or a quality
Necessity	A qualitative/subjective/existential Necessity of an object/property (e.g., should, should have, shall be, etc.)
Part-of	Being a part of an object/property
Not-part-of	Not being a part of an object/property
Greater	A value that should be greater than to
Greater-equal	A value that should be greater than or equal to
Equal	A value that should be equal to
Less-equal	A value that should be less than or equal to
Less	A value that should be less than to
None	No relation

References

Baldini Soares, L., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2895–2905). Florence, Italy: Association for Computational Linguistics.

Beach, T., Rezgui, Y., Li, H., & Kasim, T. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Systems with Applications*, *42*(12), 5219–5231.

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International conference on learning representations*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8440–8451). Online: Association for Computational Linguistics.

Costa, G., Vakaj, E., Beach, T., Lavikka, R., Lefrançois, M., Zimmermann, A., Mecharnia, T., Alexiev, V., Dridi, A., Hettiarachchi, H., & Keberle, N. (2024). Formalization of building codes and regulations in knowledge graphs. In *Digital building permit conference 2024* (pp. 120–125).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics.

Ding, Y., Ma, J., & Luo, X. (2022). Applications of natural language processing in construction. *Automation in Construction*, *136*, Article 104169.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Findings of the association for computational linguistics: ACL-IJCNLP 2021* (pp. 968–988). Online: Association for Computational Linguistics.

Fuchs, S., & Amor, R. (2021). Natural language processing for building code interpretation: A systematic literature review. In *Proceedings of the conference CIB w78: vol. 2021*, (pp. 11–15).

Fuchs, S., Witbrock, M., Dimyadi, J., & Amor, R. (2022). Neural semantic parsing of building regulations for compliance checking. *IOP Conference Series: Earth and Environmental Science*, *1101*(9), Article 092022.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8342–8360). Online: Association for Computational Linguistics.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., Han, W., Huang, M., Jin, Q., Lan, Y., Liu, Y., Liu, Z., Lu, Z., Qiu, X., Song, R., ... Zhu, J. (2021). Pre-trained models: Past, present and future. *AI Open*, *2*, 225–250.

Hassan, A., & Mahmood, A. (2018). Convolutional recurrent deep learning model for sentence classification. *IEEE Access*, *6*, 13949–13957.

Hettiarachchi, H., Adedoyin-Olowe, M., Bhogal, J., & Gaber, M. M. (2021). DAAI at CASE 2021 task 1: Transformer-based multilingual socio-political and crisis event detection. In *Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021)* (pp. 120–130). Online: Association for Computational Linguistics.

Hettiarachchi, H., Adedoyin-Olowe, M., Bhogal, J., & Gaber, M. M. (2023). TTL: transformer-based two-phase transfer learning for cross-lingual news event detection. *International Journal of Machine Learning and Cybernetics*, *14*(8), 2739–2760.

Hettiarachchi, H., Dridi, A., Gaber, M. M., Parsafard, P., Bocaneala, N., Breitenfelder, K., Costa, G., Hedblom, M., Juganaru-Mathieu, M., Mecharnia, T., Park, S., Tan, H., Tawil, A.-R. H., & Vakaj, E. (2025). CODE-ACCORD: A corpus of building regulatory data for rule generation towards automatic compliance checking. *Scientific Data*, *12*(1), 170.

Hettiarachchi, H., & Ranasinghe, T. (2023). Explainable event detection with event trigger identification as rationale extraction. In R. Mitkov, & G. Angelova (Eds.), *Proceedings of the 14th international conference on recent advances in natural language processing* (pp. 507–518). Varna, Bulgaria: INCOMA Ltd., Shoumen, Bulgaria.

Hjelseth, E., & Nisbet, N. (2011). Capturing normative constraints by use of the semantic mark-up RASE methodology. In *Proceedings of CIB w78-w102 conference* (pp. 1–10).

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8018–8025.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. In M. Johnson, B. Roark, & A. Nenkova (Eds.), *"Transactions of the Association for Computational Linguistics"*, *8*, 64–77.

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc..

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *International conference on learning representations*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Longpre, S., Wang, Y., & DuBois, C. (2020). How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 4401–4411).

Luo, L., Lai, P.-T., Wei, C.-H., Arighi, C. N., & Lu, Z. (2022). BioRED: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, *23*(5), bbac282.

Ma, Y., Wang, A., & Okazaki, N. (2023). DREEAM: Guiding attention with evidence for improving document-level relation extraction. In A. Vlachos, & I. Augenstein (Eds.), *Proceedings of the 17th conference of the European chapter of the association for computational linguistics* (pp. 1971–1983). Dubrovnik, Croatia: Association for Computational Linguistics.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. USA: Cambridge University Press.

Merchant, A., Rahimtoroghi, E., Pavlick, E., & Tenney, I. (2020). What happens to BERT embeddings during fine-tuning? In *Proceedings of the third blackboxNLP workshop on analyzing and interpreting neural networks for NLP* (pp. 33–44). Online: Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Y. Bengio, & Y. LeCun (Eds.), *1st international conference on learning representations, ICLR 2013, scottsdale, arizona, USA, May 2-4, 2013, workshop track proceedings*.

Moon, S., Lee, G., Chi, S., & Oh, H. (2021). Automated construction specification review with named entity recognition using natural language processing. *Journal of Construction Engineering and Management*, *147*(1), Article 04020147.

Nakayama, H. (2018). seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Nassiri, K., & Akhloufi, M. (2023). Transformer models used for text-based question answering systems. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(9), 10602–10635.

Obadinma, S., Guo, H., & Zhu, X. (2023). Effectiveness of data augmentation for parameter efficient tuning with limited data. In B. Can, M. Mozes, S. Cahyawijaya, N. Saphra, N. Kassner, S. Ravfogel, A. Ravichander, C. Zhao, I. Augenstein, A. Rogers, K. Cho, E. Grefenstette, & L. Voita (Eds.), *Proceedings of the 8th workshop on representation learning for NLP (repL4NLP 2023)* (pp. 226–237). Toronto, Canada: Association for Computational Linguistics.

OpenAI (2023). GPT-4 technical report. ArXiv, abs/2303.08774.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd international conference on neural information processing systems*. Red Hook, NY, USA: Curran Associates Inc..

Piskorski, J., & Yangarber, R. (2013). Information extraction: Past, present and future. In *Multi-source, multilingual information extraction and summarization* (pp. 23–49). Springer.

Plum, A., Ranasinghe, T., Jones, S., Orasan, C., & Mitkov, R. (2022). Biographical semi-supervised relation extraction dataset. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 3121–3130). New York, NY, USA: Association for Computing Machinery.

Ramshaw, L., & Marcus, M. (1995). Text chunking using transformation-based learning. In *Third workshop on very large corpora*.

Ranasinghe, T., Anuradha, I., Premasiri, D., Silva, K., Hettiarachchi, H., Uyangodage, L., & Zampieri, M. (2025). SOLD: Sinhala offensive language dataset. *Language Resources and Evaluation*, *59*(1), 297–337.

Ranasinghe, T., Orasan, C., & Mitkov, R. (2020). TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th international conference on computational linguistics* (pp. 5070–5081). Barcelona, Spain (Online): International Committee on Computational Linguistics.

Schönfelder, P., & König, M. (2021). Deep learning-based entity recognition in construction regulatory documents. In *ISARC. proceedings of the international symposium on automation and robotics in construction*: *vol. 38*, (pp. 387–394). IAARC Publications.

Shanbhogue, A. V. K., Xue, R., Saha, S., Zhang, D., & Ganesan, A. (2023). Improving low resource speech translation with data augmentation and ensemble strategies. In E. Salesky, M. Federico, & M. Carpuat (Eds.), *Proceedings of the 20th international conference on spoken language translation (IWSLT 2023)* (pp. 241–250). Toronto, Canada (in-person and online): Association for Computational Linguistics.

Shen, Y., Song, K., Tan, X., Li, D., Lu, W., & Zhuang, Y. (2023). DiffusionNER: Boundary diffusion for named entity recognition. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 3875–3890). Toronto, Canada: Association for Computational Linguistics.

Shen, Y., Tan, Z., Wu, S., Zhang, W., Zhang, R., Xi, Y., Lu, W., & Zhuang, Y. (2023). PromptNER: Prompt locating and typing for named entity recognition. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 12492–12507). Toronto, Canada: Association for Computational Linguistics.

Shen, Y., Wang, X., Tan, Z., Xu, G., Xie, P., Huang, F., Lu, W., & Zhuang, Y. (2022). Parallel instance query network for named entity recognition. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 947–961). Dublin, Ireland: Association for Computational Linguistics.

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. *Journal of Big Data*, *8*, 1–34.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*: *vol. 30*, Curran Associates, Inc..

Wang, X., & El-Gohary, N. (2021). Deep learning-based named entity recognition from construction safety regulations for automated field compliance checking. In *Computing in civil engineering 2021* (pp. 164–171).

Wang, X., & El-Gohary, N. (2023). Deep learning-based relation extraction and knowledge graph-based representation of construction safety requirements. *Automation in Construction*, *147*, Article 104696.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., ... Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45). Online: Association for Computational Linguistics.

Xue, X., & Zhang, J. (2020). Building codes part-of-speech tagging performance improvement by error-driven transformational rules. *Journal of Computing in Civil Engineering*, *34*(5), Article 04020035.

Xue, X., & Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, *47*, Article 101235.

Xue, X., & Zhang, J. (2022). Regulatory information transformation ruleset expansion to support automated building code compliance checking. *Automation in Construction*, *138*, Article 104230.

Yang, S., Choi, M., Cho, Y., & Choo, J. (2023). HistRED: A historical document-level relation extraction dataset. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 3207–3224). Toronto, Canada: Association for Computational Linguistics.

Yang, X., del Rey Castillo, E., Zou, Y., & Wotherspoon, L. (2023). Semantic segmentation of bridge point clouds with a synthetic data augmentation strategy and graph-structured deep metric learning. *Automation in Construction*, *150*, Article 104838.

Zampieri, M., Morgan, S., North, K., Ranasinghe, T., Simmmons, A., Khandelwal, P., Rosenthal, S., & Nakov, P. (2023). Target-based offensive language identification. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 2: short papers)* (pp. 762–770). Toronto, Canada: Association for Computational Linguistics.

Zhai, R., Zou, J., He, Y., & Meng, L. (2022). BIM-driven data augmentation method for semantic segmentation in superpoint-based deep learning network. *Automation in Construction*, *140*, Article 104373.

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering*, *30*(2), Article 04015014.

Zhang, J., & El-Gohary, N. M. (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction*, *73*, 45–57.

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*, Article 103834.

Zhang, Z., Ma, L., & Nisbet, N. (2023). Unpacking ambiguity in building requirements to support automated compliance checking. *Journal of Management in Engineering*, *39*(5), Article 04023033.

Zheng, Z., Lu, X.-Z., Chen, K.-Y., Zhou, Y.-C., & Lin, J.-R. (2022). Pretrained domain-specific language model for natural language processing tasks in the AEC domain. *Computers in Industry*, *142*, Article 103733.

Zheng, Z., Zhou, Y.-C., Chen, K.-Y., Lu, X.-Z., She, Z.-T., & Lin, J.-R. (2024). A text classification-based approach for evaluating and enhancing the machine interpretability of building codes. *Engineering Applications of Artificial Intelligence*, *127*, Article 107207.

Zheng, Z., Zhou, Y.-C., Lu, X.-Z., & Lin, J.-R. (2022). Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. *Automation in Construction*, *142*, Article 104524.

Zhou, P., & El-Gohary, N. (2017). Ontology-based automated information extraction from building energy conservation codes. *Automation in Construction*, *74*, 103–117.

Zhou, Y.-C., Zheng, Z., Lin, J.-R., & Lu, X.-Z. (2022). Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry*, *142*, Article 103746.