Natural Language Processing for Building Code Interpretation: A Systematic Literature Review

Stefan Fuchs, sffc348@aucklanduni.ac.nz
School of Computer Science, The University of Auckland, Auckland, New Zealand

Robert Amor, trebor@cs.auckland.ac.nz
School of Computer Science, The University of Auckland, Auckland, New Zealand

Abstract
Building codes enforce a minimum quality level for buildings to ensure the safety of building occupants. Automated code compliance checking (ACCC) can guarantee the consistent application of all relevant building codes to a building model. Recent developments in natural language processing (NLP) constitute a promising solution for automated building code computerisation (ABCC) to make them accessible for ACCC. This systematic literature review assesses the state-of-the-art of NLP for ABCC by analysing 41 research articles published since 2000. The NLP tasks range from document processing and text classification to information extraction and alignment. We categorise the studies by NLP task, arrange them into an NLP supported ACCC process and draw comparisons regarding the general characteristics, technologies used, and results and limitations. Overall, eight research gaps are identified, and recommendations for future research are provided.

Keywords: Building Codes, NLP, Deep Learning, Information Extraction, Automated Compliance Checking, Systematic Literature Review

1 Introduction

Whenever a building is constructed, altered, or demolished, a building consent is required. In New Zealand, there are over 600 codes and standards to be considered when consenting (Standards New Zealand 2021). Conventionally, getting building consent is a manual process. The authorities use checklists to ensure that all relevant requirements are fulfilled (Ministry of Business, Innovation and Employment 2014). This process can require multiple iterations until all obligations are met, consuming a significant amount of money and time (Preidel & Borrmann 2018). In the last 50 years, much commercial and academic effort has been applied to automating the compliance checking process. Eastman et al. (2009) divided the process into four steps: 1) Interpret and formalise legal requirements, 2) Extract and enrich building information, 3) Execute checks (e.g. calculations, simulations), and 4) Generate compliance reports. ACCC enables architects and project managers to precheck their design for compliance, helps consenting authorities avoid repetitive tasks, ensures consistency, and prevents errors. Most ACCC tools are facing two main challenges. The BIM does not provide sufficient compliance information of the necessary quality level, and the normative requirements, distributed over numerous codes and standards, need to be computerised and maintained to circumvent the limitation of hard-coded and potentially outdated subsets of applicable rules (Amor & Dimyadi 2021).

Regulatory documents are typically authored in natural language, intended for human interpretation. The manual translation of all building-related standards, each containing hundreds of rules, is costly and time-consuming. Due to the high complexity and domain-specific terminology, it is hard to ensure the quality and consistency of human encoded translations. Since standards are frequently amended, it is a complex chore to keep a digital version up to date especially without direct connection to the original text.

J. Zhang & El-Gohary (2017) introduced one of the first ACCC systems relying entirely on NLP for ABCC. NLP is a field in computer science that aims to process and understand human language computationally. It comprises low-level tasks like sentence tokenisation, part-of-speech (POS) tagging, and dependency parsing, as well as high-level tasks like text classification, information extraction, question answering, and machine translation. Rule-based NLP was first reported in the 1950s and is still used for domains with a lack of training data. In the 1980s, statistical methods and machine learning (ML) gained interest as computational power increased and more labelled data sets became available. The rising popularity of deep learning and large transformer-based language models like BERT (Devlin et al. 2018) and GPT3 (Brown et al. 2020) has led to incredible progress in the field. Such models show semantic and syntactic fluency, have basic world knowledge and adapt to various tasks. By building on top of a language model, we hypothesise that the complexity of domain-specific regulations can be addressed with a practical amount of resources. To the best of our knowledge, there is no survey on NLP approaches for ABCC. To fill this gap, we conduct a systematic literature review (SLR) to identify how NLP can support or automate ABCC. In the following sections, we describe the methodology, present and analyse the identified literature, discuss the gaps and suggest research directions.

2 Methodology

We adapted the SLR guidelines in Kitchenham (2004) and split the process into four parts: 1) Preparation, 2) Literature retrieval and selection, 3) Literature analysis, and 4) Documentation.

2.1 Preparation

An initial unstructured literature review was performed to identify the general interest in NLP approaches in the construction domain and construct the following research questions:

1. How can NLP technologies support or automate the interpretation of building regulations?
2. How well did varying technologies perform the interpretation tasks?
3. What level of automation can be achieved for the semantic computerisation of building codes?

The topic of this SLR is highly interdisciplinary, situated at the intersection between computer science, construction, and law. A broad selection of databases and academic search engines helped to cover these disciplines (i.e. ASCE Library, Engineering Village, Scopus, SpringerLink, ProQuest, and Google Scholar). Table 1 presents the selected search terms. Moreover, inclusion and exclusion criteria were defined to allow an objective literature selection. The application of NLP to non-normative construction documents and out-of-domain legal documents, manual building code computerisation, and information retrieval and comparison on a document level were excluded. The criteria are specified in detail in Fuchs (2021).

Table 1. Search query: "NLP terms" AND ("Building regulation terms" OR ("AEC industry terms" AND "Regulation terms"); the plural of each building regulation term was added.

NLP terms	Building regulation terms	AEC industry terms	Regulation terms
process* NEAR "natural language"	"building code"	"AEC industry"	regulation
"natural language understanding"	"building standard"	"construction industry"	regulatory
NLP	"construction code"	"building industry"	
"semantic-based"	"building regulation"	"AEC domain"	
"text analysis"	"construction regulation"	"construction domain"	
"text processing"		"building domain"	
"information extraction"		"AEC sector"	
"information retrieval"		"construction sector"	
"text classification"		"building sector"	
		"civil engineering"	

2.2 Literature retrieval and selection

Three different search strategies helped to determine the primary studies of this review. Most studies were identified in an extensive database search. Second, the citations in the background sections of the included papers were evaluated. Third, a literature search for authors with at least three included articles complements the results. We searched for English language conference

and journal articles published between 2000 and 27 April 2020 for database search and 12 August 2020 for author snowballing. The 1,962 initial database records were narrowed down by removing duplicates (1,138 remaining), screening based on titles (517 remaining) and abstracts (81 remaining), evaluating full texts (49 remaining), and removing identical studies (34 remaining). Two additional relevant papers were identified with backwards search and five papers with author snowballing, resulting in 41 articles.

2.3 Literature analysis process

First, we extracted the primary characteristics and contributions of all papers to identify the areas of interest. These key phrases were clustered based on the research questions. Table 2 shows the resulting categories, which directed the systematic extraction of information from the papers.

Table 2. Categories for the systematic analysis of the literature.

General	Technology	Results
NLP tasks	Technology type	Evaluation results
Document type	Process steps	Dataset size
Context	Technology stack	Dataset creation
Level of automation	Extracted information types	Error sources
	Representation format	Limitations
	Used features	Contributions to the field
	Domain knowledge	

2.4 Documentation

Finally, the review process and results are described and discussed in a technical report (Fuchs 2021) and summarised in this paper. In the following sections, we provide the review results and a comprehensive discussion identifying research gaps and recommendations for future research directions. The readers are referred to the technical report for full details.

3 Literature analysis

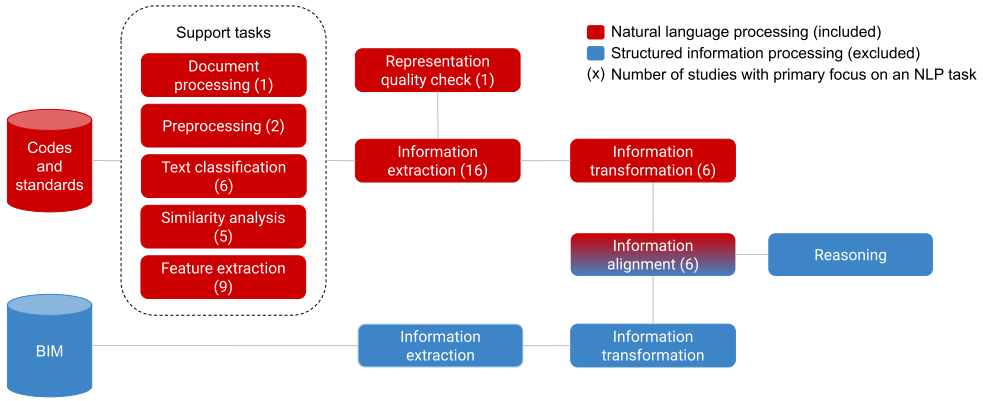

Figure 1. NLP supported automated code compliance checking process.

Before 2010, NLP was mostly used for similarity-based regulation clause retrieval. Regulations were usually transformed manually, and the research focussed on practical representation formats. This interest goes back to Fenves (1966) who used decision tables to encode design requirements. The research interest reached a peak in 2016 and has remained high since then. Over time, the technologies progressed from feature-based algorithms to ontologies to machine and deep learning. Figure 1 shows the research contributions in the various tasks of an NLP-based ACCC process. Each of these areas will be detailed in the following subsections.

3.1 Document processing

In this context, document processing refers to parsing digital regulatory documents by performing actions like de-hyphenation, removing line breaks and footnotes, and dividing the

document into sections. Most NLP-based ABCC approaches like J. Zhang & El-Gohary (2017) and Zhou & El-Gohary (2018b) used a set of regulation clauses to create their ground truth. They collected these clauses manually (Zhou & El-Gohary 2016b) or with simple algorithms applied to regulatory documents (Salama & El-Gohary 2016, Zhou & El-Gohary 2018b). Lau & Law (2004) was the only study to focus on this task. They developed a parser to transform regulatory documents from HTML, PDF, or plain text into an XML format and augmented the XML regulation clauses with features like references, concepts, and exceptions. The XML structure preserves the inherent hierarchy of the regulations and improves accessibility for subsequent processing.

3.2 Preprocessing

Preprocessing prepares the input text for NLP models or algorithms. Stanford CoreNLP (Manning et al. 2014), GATE (Cunningham et al. 2013), and NLTK (Bird et al. 2009) were commonly used for sentence splitting, tokenisation, morphological analysis, and the removal of stop words and rare words. Domain-specific preprocessing was used to deal with regulation specific traits. While Al Qady & Kandil (2010) split regulation clauses containing lists manually, Zhou & El-Gohary (2017) proposed a technique to remove quotation marks and text in parenthesis, split exceptions, conjunctions, and lists from the clauses, and stitch the corresponding heading and relationship indicators. The tokenisation of languages without word boundaries is more complex. Common approaches are based on rules, statistics, or dictionaries. J. Zhang et al. (2018) tokenised Chinese building specifications using a hashed dictionary and a reverse maximum matching algorithm.

3.3 Text classification

Text classification techniques can be used to filter out irrelevant regulation clauses. Zhou & El-Gohary (2016a,b) classified the clauses into environmental topics, Song et al. (2018) and Salama & El-Gohary (2016) into general ACCC categories, and Le et al. (2019) and Hassan & Le (2020) into requirements and non-requirements. Machine learning (ML) was the prevalent technology with a wide range of experiments being performed to identify the best algorithms and features for the tasks. Three studies identified support vector machines as the best suited ML-algorithm. To avoid feature engineering, Song et al. (2018) used deep learning and Zhou & El-Gohary (2016b) calculated the similarities between the vector representations of clauses and ontology concepts. Most of the authors agreed on a 100% recall target since falsely classified clauses can cause the system to miss non-compliant building specifications. Salama & El-Gohary (2016) showed that this goal is achievable for binary text classification tasks.

3.4 Similarity analysis

A common technique for clustering and information retrieval is to evaluate the similarity of text based on word stems or vector representations. Lau & Law (2004) and Lau et al. (2006) used the feature-enriched XML repository described in Section 3.1 to calculate the similarity between regulation clauses and refined the similarity scores using parent and sibling clauses and references. Cheng et al. (2008) leveraged the relatedness analysis of Lau et al. (2006) for taxonomy-based regulation retrieval. Song et al. (2018) introduced an expert support system for ABCC. While transforming a requirement, the system shows related regulations, classifies the clause by topic, and allows the user to search for nouns and predicates related to the entered search term. R. Zhang & El-Gohary (2018) used unsupervised ML to cluster building codes to determine differences in complexity and structure, and consequently, in computability. They identified seven requirement types, which vary in sentence length, the number of independent and dependent clauses, missing essential information, and the existence of restrictions. About 60% of the sentences accounted for simple requirements with high to medium computability.

3.5 Feature extraction

Semantic and syntactic text analysis can be used to determine linguistic features of a text. Structural features like POS tags, phrase chunks, and dependency trees were commonly used for rule-based information extraction (IE). J. Zhang & El-Gohary (2012) compared the suitability of phrase structure grammar and dependency grammar for feature-based algorithms. R. Zhang & El-Gohary (2019c) developed a neural network dependency parser (DP) optimised for building

codes to generate features for a rule-based requirement unit extraction. Using the custom DP instead of the Stanford DP accounted for an overall improvement of 2% for the requirement extraction. Wrong POS tags were a common error source in rule-based IE (J. Zhang & El-Gohary 2016b, Zhou & El-Gohary 2017). Xue & Zhang (2020b) evaluated seven POS taggers on building codes and identified word ambiguities, rare words, and unique word meanings as leading causes for wrong POS tags. Xue & Zhang (2020a) aimed to resolve this issue with an error correction algorithm to fix common building code POS tag errors. They were able to enhance the accuracy from 89.4% to 98.1%. A method to create semantic features is semantic role labelling (SRL), an area with large established data sets like FrameNet and PropBank. SRL aims to label a sentence with roles like agents (i.e. the acting entity), recipients (i.e. the target of an action), actions, and various modifiers (e.g. adverb, location, manner, and temporal). Those roles are related to the information types required for ACCC. R. Zhang & El-Gohary (2019a,d) labelled building codes with semantic roles and used these labels and syntactic features to automatically identify IE templates.

3.6 Information extraction (IE)

IE is a generic term for the identification of semantic information elements (SIE), requirement units, events, and relations in unstructured text. IE was the common task among all ABCC approaches. The studies vary substantially in the depth of IE. The differences range from extracting concepts and relations (Al Qady & Kandil 2010, Fahad et al. 2016, Shi & Roman 2017) to the minimum SIEs sufficient to represent simple requirements like building element, property, quantity, relation, and function (e.g. Kwon et al. (2013), Niemeijer et al. (2014), Emani et al. (2016)), to an extensive IE of around ten SIEs for complex requirements (e.g. J. Zhang & El-Gohary (2016b), Zhou & El-Gohary (2017), Xu & Cai (2019)). Figure 2 shows a simple example annotated with the SIEs defined in J. Zhang & El-Gohary (2016b). Li et al. (2016) and Xu et al. (2019) used SIEs adapted for spatial constraints prevalent in utility regulations. The restrictions contained in complex requirements were either extracted directly (e.g. subject and quantity restrictions (J. Zhang & El-Gohary 2016b)), determined as a composition of SIEs (Zhou & El-Gohary 2017), or the clauses were split into requirement units first (R. Zhang & El-Gohary 2019c), and these requirement units were the input for further IE (R. Zhang & El-Gohary 2020b).

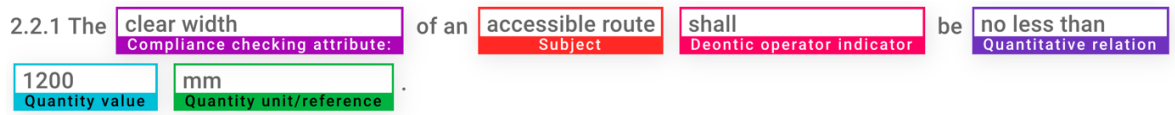

Figure 2. Simple regulation clause 2.2.1 from Ministry of Business, Innovation and Employment (2014) annotated with the SIEs from J. Zhang & El-Gohary (2016b). Annotated with doccano (Nakayama et al. 2018).

An early strategy for IE combined rules and features created by NLP tools (e.g. POS tags and dependency trees). Most rule-based approaches used gazetteer lists (i.e. ensembles of fixed terms) to extract some of the SIEs. These lists are well suited for SIEs with little variation like negations, quantity units, and comparative relations. Although gazetteers were also used to capture domain knowledge (Kwon et al. 2013, Li et al. 2016, Xu et al. 2019), the use of ontologies was the prevalent technique to represent domain concepts and their relations (e.g. Kwon et al. (2013), Mathot et al. (2016), J. Zhang & El-Gohary, (2016b), Zhou & El-Gohary (2017)). Ontologies have the advantages of higher reusability and information density. Xu & Cai (2019) combined semantic and syntactic features by using frames. Since they tested only one frame for IE with 92.3% precision, it is unclear how well the frame-based IE performs with multiple frames.

Starting in 2019, researchers applied deep learning to the IE task to address the scalability limitation arising from rule- and ontology-based approaches. For Moon et al. (2019) and Zhong et al. (2020), a lack of training data caused insufficient results (i.e. 25.6% and 73.7% F-measure). R. Zhang & El-Gohary (2020b) used a bidirectional LSTM model (Hochreiter & Schmidhuber 1997) in combination with transfer learning strategies to address the lack of training data. Transfer learning denotes using out-of-domain training data or pretrained models and refining them for the actual task. With an F-measure of 87%, they showed the potential of this method and outperformed Moon et al. (2019) and Zhong et al. (2020), but they are far from the best rule-

and ontology-based approaches (i.e. 95.6% and 97.9% F-measure in J. Zhang & El-Gohary, (2016b) and Zhou & El-Gohary (2017), respectively).

3.7 Information transformation

After IE, the extracted information can be postprocessed and transformed into intermediate formats (e.g. information tuples (Li et al. 2016, J. Zhang & El-Gohary 2016b, Zhou & El-Gohary 2017), regulation trees (Niemeijer et al. 2014), SWRL (Fahad et al. 2016, Shi & Roman 2017), mvdXML (Fahad et al. 2016), RAINS (Emani et al. 2016), deontic logic (Xu et al. 2019)) and further into executable representations (e.g. SPARQL (Emani et al. 2016), XSLT (Shi & Roman 2017), Prolog logic rules (J. Zhang & El-Gohary 2015, Zhou & El-Gohary 2018b), PL/SQL (Xu et al. 2019)). The intermediate formats are usually closer to the original regulations and easier to read by humans. Several review papers compare the suitability of representation formats for ACCC (Nawari & Alsaffar 2015, BuildingSMART 2017, Solihin et al. 2019). With a sufficiently deep IE, the transformation can be automated using rules (e.g. Niemeijer et al. 2014, Emani et al. 2016, Li et al. 2016, Xu et al. 2019, J. Zhang & El-Gohary 2015). The identification of missing or duplicated semantic information elements (SIE) was commonly part of this step. The complexity of the rules grows with the number of SIEs. J. Zhang & El-Gohary (2013, 2015) experimented with strategies to deal with this complexity. J. Zhang & El-Gohary (2013) suggests a bottom-up approach, where the clauses annotated with SIEs and syntactic features are traversed and matched against a set of patterns. J. Zhang & El-Gohary (2015) performed further experiments using the bottom-up approach. First, they extracted the eight essential SIEs (i.e. no restrictions or exceptions) and transformed them into 1,114 logic clause elements (93.8% F-Measure). Second, to extract restrictions and exceptions, they added syntactic and combinatorial information tags (i.e. 40 information tags). The number of semantic mapping rules increased by 460% to 297. The higher information density allowed for F-Measure improvements to 98.6%.

3.8 Information alignment

With the progression towards executable formats, there is a need to align the information originating from building codes with the information from BIM or geographic information systems in utility compliance checking. J. Zhang & El-Gohary (2016a) used term-based matching and utilised WordNet (Princeton University 2010) to be able to match synonyms. Zhou & El-Gohary (2018a) looked up concepts and properties in ontologies and the buildingSMART Data Dictionary (bSDD) (BuildingSMART 2021) and identified the final match by comparing the similarity scores (98.0% recall and 89.2% precision). R. Zhang & El-Gohary (2019b) concatenated general word embeddings with domain word embeddings to encode and compare the concepts that should be aligned. Additionally, they used supervised ML to align relations like spatial composition, material constituent, and property and achieved an accuracy of 77.5%.

3.9 NLP-based automated code compliance checking (ACCC)

NLP-based ACCC systems rely on NLP to automatically retrieve, interpret, and align regulatory and design information. The processed information serves as input to reason about building compliance. J. Zhang & El-Gohary (2017) integrated information extraction, transformation, and alignment into a unified ACCC system. Zhou & El-Gohary (2018b) added a text classification step to their system to filter for relevant regulations. While most of the tasks were explained in greater details in the task-specific papers, these papers contributed an end-to-end evaluation. J. Zhang & El-Gohary (2017) achieved an F-measure of 92.8% in finding 79 non-compliant instances in a building. The regulation information extraction and transformation were the primary error sources. Zhou & El-Gohary (2018b) achieved 88% F-Measure to extract 24 non-compliant instances with information alignment as the main error source. The differences can be explained by the higher number of restrictions in energy codes and the replacement of transformational alignment rules with an explicit information alignment step in Zhou & El-Gohary (2018b).

3.10 Quality assurance

The quality of the digital representations is of high importance since the ACCC frameworks need a solid foundation to find acceptance. While most studies have developed gold standards to

evaluate their approaches, these data sets varied widely in size and quality. For example, R. Zhang & El-Gohary (2020b) evaluated their semantic annotations with 30 sentences, Li et al. (2016) used 30 simple and 20 complex clauses to test both information extraction and transformation, and Zhou & El-Gohary (2017) used one energy code chapter to test the IE of 659 instances. In many cases, there were no details about the labelling process (Kwon et al. 2013, Li et al. 2016, Xu & Cai 2019, R. Zhang & El-Gohary 2020b), some studies had one annotator and multiple reviewers (J. Zhang & El-Gohary 2015, 2016b, Zhou & El-Gohary 2016b), and Zhou & El-Gohary (2017) had three annotators aiming for full annotator agreement. J. Zhang & El-Gohary (2016a) added a manual review step to assure the quality of the extracted regulation concepts. R. Zhang & El-Gohary (2020a) leveraged natural language generation (NLG) to recreate building code sentences from the extracted SIEs. The NLG metrics BLEU (Papineni et al. 2002) and ROGUE (Lin 2004) were used to evaluate the quality of the SIEs. Both metrics measure the overlap of n-grams. Since BLEU is precision-based and ROUGE is recall-based, these metrics complement each other well. The achieved scores between 73% and 86% were interpreted as good comprehensibility.

4 Gaps in research

The data extracted and categorised based on Table 2 was analysed to identify eight research gaps described in the following subsections. Underrepresented tasks, differences in the extracted information types and representation formats, the utilised domain knowledge, and finally, the evaluation results, limitations, and error sources are the primarily reflected gap categories.

Gap 1: Insufficient regulation context

Current approaches do not take advantage of the full context of clauses, instead focusing on individual clauses as standalone entities. Consequently, the connection to the original document structure is lost, and relevant information like definitions, instructions on how to apply the regulations, and restrictions inherited from parent provisions are neglected. First, the document structure and non-requirement text should be preserved in the representation format to improve the accessibility of this information (e.g. LegalDocML and LegalRuleML in Dimyadi et al. (2020)). Subsequently, this contextual knowledge could be utilised for the semantic interpretation.

Gap 2: No public data sets

Currently, there are no public benchmark data sets for the information extraction from building codes. The only benchmark data sets are for text classification (Zhou & El-Gohary 2016a, Hassan & Le 2020) and POS tagging (Xue & Zhang 2020b). Since most studies vary in test data, extracted SIEs, and representation formats, a direct comparison is not possible. Many of the test sets were relatively small and without meta-data about labelling processes. A trustworthy, diverse, accepted, and open data set could enhance comparability and competition among researchers worldwide and allow research teams to progress faster. Therefore, the data set should be quality assured and cover different normative texts (e.g. codes, standards) from multiple jurisdictions, reflecting the differences between performance-based and prescriptive building codes, and the complexity of clauses should be balanced (R. Zhang & El-Gohary 2018).

Gap 3: Agreement on complete representation requirements

The examined studies do not agree about the information required to fully represent a regulation and enable ACCC. Most studies differed in the extracted semantic information, and the representations used for NLP-based ACCC were often specialised for quantitative or spatial requirements. BuildingSMART has a working group addressing this issue, which requires international consensus. BuildingSMART (2017) identified the interoperability between formats, missing world knowledge, representing conjunctive and disjunctive relations duplication free, dealing with uncertainty, and incorporating checking methods as the main technical issues.

Gap 4: Enabling scalable information extraction with exceptional performance

More research is required to enable scalable and high-performing deep IE. The frequently used ontologies and rules were developed manually or semi-automatically and specialised to sub-domains. Unknown terms (Li et al. 2016), implicit knowledge (Zhou & El-Gohary 2017, Xu et al. 2019), missing rules (Li et al. 2016, J. Zhang & El-Gohary 2016b, Zhou & El-Gohary 2017), complex sentence structures (Li et al. 2016, J. Zhang & El-Gohary 2016b), and errors made by NLP tools (Li et al. 2016, J. Zhang & El-Gohary 2016b, Zhou & El-Gohary 2017) were identified as common

error sources. One way of dealing with these limitations was to implement deep learning-based IE, but the performance of these approaches has not yet caught up with rule-based methods. Much of the recent successes in NLP can be affiliated with large transformer-based models like BERT (Prasanna et al. 2020). Consequently, we expect improved IE results by leveraging transformer-based architectures and pretrained language models (Nguyen et al. 2020).

Gap 5: Enabling scalable information alignment with exceptional performance

Scalability and ambiguity issues with ontology-based methods, and low performance and a limited selection of IFC concepts in the machine learning-based approach cause demand for further research on information alignment. Zhou & El-Gohary (2018b) identified information alignment as the main error source in the end-to-end tests. Especially, super-concepts and restrictions were challenging to identify. We suggest pretrained models refined on domain text and combined with structured domain knowledge (e.g. bSDD) for future development.

Gap 6: Expanding beyond quantitative textual requirements

Only a few researchers covered existential and qualitative requirements, and no study could deal with tables and figures in codes and standards. Although tables have the advantage of being in a structured form, they are often highly nested and complex. For a reliable transformation, one needs to take the corresponding provision, the table caption, the headers, the entry formats, and more into account. Finally, the interpretation of figures represents the most challenging problem since they contain textual and visual information. More evaluation is required to determine whether an automated or semi-automated process is viable or exceeds the cost-benefit ratio.

Gap 7: Incorporating complex requirements

Most of the IE approaches could not deal with the entire complexity of regulations (e.g. restrictions, conjunctions, exceptions, lists, cross-references, etc.). Splitting exceptions, lists, and other conjunctions into separate clauses (Zhou & El-Gohary 2017) and breaking down the regulations into requirement units to comprehensively interpret restrictions and identify their relationships (R. Zhang & El-Gohary 2019c) encompass the scope of this task. Especially, the non-quantitative characteristic of most restrictions needs more consideration in the future.

Gap 8: Standardising quality assurance

Besides using test sets, the manual review of extracted concepts, and the generation of the original text, there was no research on quality assuring the transformed regulations. Since the performance requirements for the digitalisation of regulations are exceptionally high, it is unknown whether an NLP approach can ever achieve a quality that will be acceptable for officials. As a large percentage of the effort lies in the nonrecurring, initial creation of a digital representation, we suggest keeping the human in the loop by integrating the code transformation with a manual review, active learning, and ultimately with the rule authoring process.

5 Conclusion

The prevalent NLP approaches surveyed here are based on structural features, ontologies, and rules. Although performing well, they are widely considered to have low scalability and high reliance on the quality of the rules and knowledge bases. Machine learning has been explored to fill the gap, but these studies have not reached the accuracy of rule-based approaches. The scarcity of training data, the lack of open data sets, and the disagreement about a suitable representation for building regulations are hindrances to rapid improvement. However, deep learning is a field of rapid improvements. New discoveries are made frequently and open up opportunities to address some of these problems. Future research should target a high-quality translation of various normative building documents, including textual, tabular, and graphical requirements. Such a translation system provides the foundation for integrating the development and maintenance of a digital regulation in the rule-authoring process.

Acknowledgements

This research was funded by the University of Canterbury's Quake Centre's Building Innovation Partnership (BIP) programme, which is jointly funded by industry and the Ministry of Business, Innovation and Employment (MBIE).

References

Al Qady, M. & Kandil, A. (2010). Concept relation extraction from construction documents using natural language processing. *Journal of Construction Engineering and Management*. 136(3). pp. 294–302.

Amor, R., & Dimyadi, J. (2020). The Promise of Automated Compliance Checking. *Developments in the Built Environment*.

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python. *O'Reilly Media, Inc*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., . . . Amodei, D. (2020). Language models are few-shot learners. *arXiv*.

BuildingSMART (2017). Regulatory Room Report on Open Standards for Regulations, Requirements and Recommendations Content. *buildingSMART Standards Summit 2017 in Barcelona(1)*. pp. 1–152.

BuildingSMART (2021). buildingSMART Data Dictionary. *https://github.com/buildingSMART/bSDD*.

Cheng, C.P., Lau, G.T., Law, K.H., Pan, J. & Jones, A. (2008). Regulation retrieval using industry specific taxonomies. *Artificial Intelligence and Law*. 16(3). pp. 277-303.

Cunningham, H., Tablan, V., Roberts, A., & Bontcheva, K. (2013). Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS Comput Biol*. 9(2). e1002854.

Devlin, J., Chang, M.W., Lee, K. & Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*. arXiv:1810.04805.

Dimyadi, J., Fernando, S., Davies, K., & Amor, R. (2020). Computerising the New Zealand Building Code for Automated Compliance Audit. *6th New Zealand Built Environment Research Symposium (NZBERS2020)*. 6. pp. 39–46.

Eastman, C., Lee, J. M., Jeong, Y. S., & Lee, J. K. (2009). Automatic rule-based checking of building designs. *Automation in construction*. 18(8). 1011-1033.

Emani, C., da Silva, C.F., Fiès, B., Zarli, A. & Ghodous, P. (2016). An Approach for Automatic Formalization of Business Rules. *Proc. of the 33rd CIB W78 Conference 2016*. pp. 11.

Fahad, M., Bus, N. & Andrieux, F. (2016). Towards mapping certification rules over BIM. *CIB W78 Conference (Vol. 3).*

Fenves, S. J. (1966). Tabular decision logic for structural design. *Journal of the Structural Division*. 92(6). pp. 473-490.

Fuchs, S (2021). Natural Language Processing for Building Code Interpretation: Systematic Literature Review Report. doi: 10.13140/RG.2.2.29107.55845

Hassan, F. U., & Le, T. (2020). Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*.

Kitchenham, B. (2004). Procedure for Performing Systematic Literature Reviews. *Keele University: Newcastle, UK.*

Kwon, J., Kim, B., Lee, S. & Kim, H. (2013). Automated procedure for extracting safety regulatory information using natural language processing techniques and ontology. *Annual conference of the canadian society for civil engineering 2013*. Vol. 2. pp. 1213– 1220.

Lau, G. T. & Law, K. H. (2004). An Information Infrastructure for Comparing Accessibility Regulations and Related Information from Multiple Sources. *Proceedings of the 10th international conference on computing in civil and building engineering*. pp. 1–11.

Lau, G. T., Law, K. H. & Wiederhold, G. (2006). A relatedness analysis of government regulations using domain knowledge and structural organization. *Information Retrieval*, 9(6). pp. 657–680.

Le, T., Le, C., Jeong, H. D., Gilbert, S. B. & Chukharev-Hudilainen, E. (2019). Requirement Text Detection from Contract Packages to Support Project Definition Determination. *Advances in informatics and computing in civil and construction engineering.*

Li, S., Cai, H. & Kamat, V. R. (2016). Integrating Natural Language Processing and Spatial Reasoning for Utility Compliance Checking. *Journal of Construction Engineering and Management*. 142(12).

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. pp. 55-60.

Mathot, M., Coenders, J., & Rolvink, A. (2016, September). Feasibility of a Knowledge-Based Engineering framework for the AEC industries*. Proceedings of IASS Annual Symposia*. Vol. 2016. 13. pp. 1-9.

Ministry of Business, Innovation and Employment (2014). New Zealand Building Code Handbook — Third edition — Amendment 13.

Moon, S., Lee, G., Chi, S. & Oh, H. (2019). Automatic review of construction specifications using natural language processing. *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*. pp. 401-407.

Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., & Liang, X. (2018). doccano: Text annotation tool for human. Retrieved from https://github.com/doccano/doccano.

Nawari, N. O. & Alsaffar, A. (2015). Understanding computable building codes. *Civil Engineering and Architecture*. 3(6).

Nguyen, M. T., Phan, V. A., Son, N. H., Hirano, M. & Hotta, H. (2019). Transfer learning for information extraction with limited data. *International Conference of the Pacific Association for Computational Linguistics*. pp. 469-482.

Niemeijer, R. A., De Vries, B. & Beetz, J. (2014). Freedom through constraints: User- oriented architectural design. *Advanced Engineering Informatics*. 28(1). pp. 28–36.

Papineni, K., Roukos, S., Ward, T. & Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. pp. 311-318.

Prasanna, S., Rogers, A., & Rumshisky, A. (2020). When bert plays the lottery, all tickets are winning. *arXiv preprint.* arXiv:2005.00561.

Preidel, C., & Borrmann, A. (2018). BIM-based code compliance checking. *Building information modeling: Technology foundations and industry practice*. pp. 367–381.

Princeton University (2010). WordNet. *Princeton University "About WordNet.".*

Salama, D. M., & El-Gohary, N. M. (2016). Semantic text classification for supporting automated compliance checking in construction. *Journal of Computing in Civil Engineering*. 30(1). 04014106.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation. 9(8). 1735-1780.

Shi, L., & Roman, D. (2017). From standards and regulations to executable rules: A case study in the Building Accessibility domain. *Ceur workshop proceedings*. Vol. 1875.

Solihin, W., Dimyadi, J., & Lee, Y.-C. (2019). In Search of Open and Practical Language-Driven BIM-Based Automated Rule Checking Systems. *Advances in informatics and computing in civil and construction engineering*.

Song, J., Kim, J., & Lee, J. K. (2018). NLP and deep learning-based analysis of building regulations to support automated rule checking system. *Proceedings of ISARC*. Vol. 35. pp. 1-7.

Standards New Zealand. (2021). Standards New Zealand - Online Library catalogues. *https://www.standards.govt.nz/get-standards/standards-access-solutions/online-library-subscriptions/online-library-catalogues/*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint*. arXiv:1706.03762.

Xu, X., & Cai, H. (2019). Semantic Frame-Based Information Extraction from Utility Regulatory Documents to Support Compliance Checking. *Advances in informatics and computing in civil and construction engineering.* pp. 223-230.

Xu, X., Cai, H., & Chen, K. (2019). Modeling 3D spatial constraints to support utility compliance checking. *Computing in civil engineering 2019: Visualization, information modeling, and simulation*. pp. 439-446.

Xue, X., & Zhang, J. (2020a). Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules. *Journal of Computing in Civil Engineering*. 34. pp. 1–10.

Xue, X., & Zhang, J. (2020b). Evaluation of Seven Part-of-Speech Taggers in Tagging Building Codes: Identifying the Best Performing Tagger and Common Sources of Errors. *Construction Research Congress 2020: Computer Applications*. pp. 498-507.

Zhang, J., Chen, Y., Hei, X., Zhu, L., Zhao, Q., & Wang, Y. (2018). A RMM based word segmentation method for Chinese design specifications of building stairs. *Proceedings - 14th international conference on computational intelligence and security, cis 2018*. pp. 277–280.

Zhang, J., & El-Gohary, N. M. (2012). Extraction of construction regulatory requirements from textual documents using natural language processing techniques. *Congress on computing in civil engineering, proceedings*. pp. 453–460.

Zhang, J., & El-Gohary, N. M. (2013). Handling sentence complexity in information extraction for automated compliance checking in construction. *Proc., CIB W78 2013*.

Zhang, J., & El-Gohary, N. M. (2015). Automated information transformation for automated regulatory compliance checking in construction. *Journal of Computing in Civil Engineering*. 29(4). B4015001.

Zhang, J., & El-Gohary, N. M. (2016a). Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques. *Journal of Computing in Civil Engineering*. 30(5).

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

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. pp. 45–57.

Zhang, R., & El-Gohary, N. M. (2018). A clustering approach for analyzing the computability of building code requirements. *In Construction Research Congress 2018*. pp. 86-95.

Zhang, R., & El-Gohary, N. (2019a). A Machine Learning Approach for Compliance Checking-Specific Semantic Role Labeling of Building Code Sentences. *Advances in informatics and computing in civil and construction engineering*. pp. 561–568.

Zhang, R., & El-Gohary, N. (2019b). A Machine-Learning Approach for Semantic Matching of Building Codes and Building Information Models (BIMs) for Supporting Automated Code Checking. *International Congress and Exhibition "Sustainable Civil Infrastructures"*. pp. 64-73.

Zhang, R., & El-Gohary, N. (2019c). A machine learning-based method for building code requirement hierarchy extraction. *2019 Canadian Society for Civil Engineering Annual Conference, CSCE 2019*.

Zhang, R., & El-Gohary, N. (2019d). Unsupervised Machine Learning for Augmented Data Analytics of Building Codes. *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*. pp. 74-81.

Zhang, R., & El-Gohary, N. (2020a). A Deep-Learning Method for Evaluating Semantically-Rich Building Code Annotations. *EG-ICE 2020 Workshop*. pp. 285–293.

Zhang, R. & El-Gohary, N. (2020b). A Machine-Learning Approach for Semantically-Enriched Building-Code Sentence Generation for Automatic Semantic Analysis. *Construction Research Congress 2020: Computer Applications.*

Zhong, B., Xing, X., Luo, H., Zhou, Q., Li, H., Rose, T., & Fang, W. (2020). Deep learning- based extraction of construction procedural constraints from construction regulations. *Advanced Engineering Informatics*. 43.

Zhou, P., & El-Gohary, N. (2016a). Domain-specific hierarchical text classification for supporting automated environmental compliance checking. *Journal of Computing in Civil Engineering*. 30(4).

Zhou, P., & El-Gohary, N. (2016b). Ontology-based multilabel text classification of construction regulatory documents. *Journal of Computing in Civil Engineering*. 30(4).

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

Zhou, P., & El-Gohary, N. (2018a). Automated matching of design information in BIM to regulatory information in energy codes. *Construction research congress 2018*. pp. 75–85.

Zhou, P., & El-Gohary, N. (2018b). Text and Information Analytics for Fully Automated Energy Code Checking. *International Congress and Exhibition "Sustainable Civil Infrastructures".* pp. 196-208.