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

Technical Report · May 2021

DOI: 10.13140/RG.2.2.29107.55845

1 author:

Stefan Fuchs
University of Auckland
2 PUBLICATIONS **1** CITATION

SEE PROFILE

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

Stefan Fuchs
sffc348@aucklanduni.ac.nz
The University of Auckland

Technical report
May 2021

Abstract

Building codes enforce minimum quality level for buildings to ensure the safety of building occupants. Automated compliance checking can guarantee the consistent application of all relevant building codes to a building model. Although automated compliance checking has received considerable research interest, there are no tools available that can automate the entire compliance checking process. Since most countries' building regulations are only available in natural language, much research effort flows into encoding these regulations in a representation that can be understood by a computer. Current deep learning Natural Language Processing (NLP) techniques can achieve a comprehensive understanding of a text and constitute a promising solution for the automated computerisation of building regulations.

This systematic literature review assesses the state-of-the-art of NLP for building code interpretation by analysing 42 research articles published since 2000. These were selected from 1,962 records retrieved from six databases, plus further candidate articles detected by backwards snowballing and author search strategies. The studies used NLP to process regulatory documents, analyse text and extract syntactic and semantic features, filter out irrelevant clauses using text classification, and extract the information required to transform the regulation into a computer-processable format. Semantic alignment of the regulation concepts and the building design information fills the gap to automated reasoning. The information extraction task received the highest research interest. The approaches are highly variant in the depth of extracted information and the complexity of the regulations. Rule- and ontology-based approaches were commonly used and reached high performance but are highly task-specific and difficult to scale. Although state-of-the-art machine learning techniques showed the potential to provide a scalable solution, there is still room for improvement.

Overall, eight research gaps were identified in the current literature. Information extraction and information alignment were the most complex and challenging tasks, where methods that are both scalable and high-performing are still missing. Also, most studies were limited to quantitative requirements. For a full digital version of building regulations, entire regulatory documents with all expressions of requirements (e.g. textual, tabular and figure encoded) need to be translated, and high quality needs to be assured.

Contents

Chapter 1

Introduction

In New Zealand, whenever a building is going to be constructed, altered, or demolished a building consent is required. In the case of a new or altered building, the building plan drawings will be passed to the authorities to be checked against relevant codes and standards. Overall, there are over 600 codes and standards to be considered when consenting (Standards New Zealand, 2021). Conventionally, getting a building consent is a manual process. Checklists are used to ensure that all relevant requirements are fulfilled. Specialists or third parties might be consulted for checks related to performance-based or engineering-specific design proposals. The consenting authorities have 20 working days to process a request once the application is complete. If further supporting information is required during the review process, the applicant will need to provide missing documents, and the application processing time is suspended (Ministry of Business, Innovation and Employment, 2014). Especially for larger projects, getting a building consent can take multiple iterations until all obligations are met, and there are no further changes to the design. Accordingly, it consumes a significant amount of money and time (Preidel & Borrmann, 2018). To accelerate this process, an automated process for compliance checking can be a valuable tool for both parties.

For architects and project managers, an Automated Code Compliance Checking (ACCC) system gives a chance to check for compliance against codes and standards in earlier stages of planning and design, and before applying for building consent. Expensive design changes could be prevented since compliance breaches can be resolved earlier or avoided altogether. The building consent application process would be much simpler without a need to specify in detail how the building complies to the building code requirements. It also allows better customisation and more innovation in building designs (Niemeijer et al., 2014). The performance-based building codes in New Zealand are designed to allow innovative design by prescribing only the end goal, without enforcing the 'how'. Most of the building code clauses have associated acceptable solutions and verification methods. These documents provide standard methods to comply with the building codes. If the acceptable solution for a relevant building code clause is implemented, the

building consent authority must accept the design regarding these requirements. However, suppose the acceptable solution does not fully cover the chosen design. In that case, the applicant must provide alternative solutions to meet the minimum expectations (e.g. calculations, simulations, comparisons with acceptable solutions or previously accepted solutions). Nonetheless, the creative freedom is often neglected, with designers following the acceptable solutions to avoid the complex process of proving compliance using alternative solutions.

For authorities, an automated tool could help to avoid repetitive tasks and leave time for assessments where human expertise is necessary. Due to the large number of codes and standards, a manual compliance checking process is prone to errors and inconsistencies. Fiatech Regulatory Streamlining Committee (2012) discovered that different consenting authorities are likely to produce inconsistent compliance checking results even if there are no differences in the applied regulations.

In recent years, the adoption of Building Information Modelling (BIM) is rising. BIM does not only stand for the digital model of a building but for a way of collaboration between different stakeholders over the entire building lifecycle (BIM Acceleration Committee, 2019). As an outcome, BIM offers a broad spectrum of information usable for compliance checks. A number of tools have already checked BIMs successfully for compliance issues. Solibri Model Checker is one of the most renown commercial tools on the market. A weakness of this tool is the limited number of hard-coded rules. Although the user can change and add rules, it is very labour intensive to do that manually for all applicable provisions. ACCC has also piqued the interest of many researchers in the last decades. Nevertheless, there are hardly any tools that transformed from the proof-of-concept to a real-world application (Solihin et al., 2019). One reason for this trend is the availability of codes and standards in natural language only. Moreover, due to these texts' complexity, there is a lack of tools to automate or support the computerisation of regulations in a feasible way.

The manual translation of over 600 building-related standards in New Zealand, each containing hundreds of rules, is a costly and time-consuming venture. Regulatory documents are typically authored in natural language, intended for human interpretation. Natural language has theoretically no limitations on complexity as one can always extend a sentence by adding another clause or phrase. The building regulations have the advantage that they are semi-structured and typically use a specialised vocabulary leading to fewer ambiguities. However, they also use legal and domain-specific terminology that requires special knowledge from the translators. Overall, it is hard to ensure the quality and consistency of human encoded translations. Since the standards are frequently amended, it is a complex chore to keep a digital version up to date especially without having a direct connection to the original text. For example, Dimyadi et al. (2020) promoted the idea of a "digital twin" of the regulations using a combination of the open standards LegalDocML and LegalRuleML.

J. Zhang & El-Gohary (2017) introduced one of the first compliance checking frameworks that entirely relied on Natural Language Processing (NLP) for the conversion of building regulations. They developed rules that used syntactic and semantic text features to extract information elements from the International Building Code (IBC) and convert the extracted elements into logic rules. They showed that there is a potential in using NLP to automate the interpretation, but they tested their approach only with quantitative requirements of one IBC chapter, and rule-based NLP is usually limited in its scalability since the rules are specialised for a particular text type (R. Zhang & El-Gohary, 2019b).

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, machine translation, and text summarisation. Rule-based NLP was first reported in the 1950s and is still in use for domains with a lack of training data. In the 1980s, statistical methods and machine learning gained interest as the computational power increased, and more labelled data sets became available. The high costs for labelling large scale training data led the research towards unsupervised or semi-supervised machine learning, which makes use of the large amount of text available on the worldwide web. The major success of unsupervised learning arose with the increasing popularity of neural networks and deep learning in recent years. These systems work without feature engineering by numerically encoding the text in high-dimensional spaces. Algorithms like word2vec (Mikolov et al., 2013) are commonly used to create meaningful representations, so-called word embeddings, from large text corpora by predicting neighbouring words or the next word in a sentence and adjusting the vector representations based on this prediction using backpropagation. This process leads to clusters of words in the vector space based on syntactic and semantic characteristics.

Such word embeddings are typically used to initialise neural networks like Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNNs are widespread in computer vision but also used for NLP tasks like text classification. A convolution is a filter that can automatically identify features in data. Numerous parallel and sequential convolutions are used to capture the various features on different detail levels. RNNs encode sentences word by word. In each step, they concatenate the word vector with the current hidden state of the sentence. The result of each calculation is the hidden state for the next calculation. To include backward and forward relationships, the bi-directional RNNs encode sentences from both directions and concatenate the hidden states for each position. A common problem of RNNs is that the signal gets weaker with distance. That means, by the end of a long sentence, the model might not remember anything about the start. Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2015) models introduce methods to

conserve crucial information over a long distance while forgetting unimportant information.

Large improvements were achieved by introducing attention mechanisms in the models (Bahdanau et al., 2015; P. Zhou et al., 2016; Seo et al., 2016). Attention allows focusing on relevant parts of a sentence while producing the output. Take the translation of the sentence "The old cat is in the house." to German "Die alte Katze ist im Haus." as an example. While translating the article "The", one must also pay attention to the noun "cat". That allows choosing the translation in the correct genus (i.e. "die").

Vaswani et al. (2017) revolutionised the field by arguing that RNNs can be completely replaced by attention mechanisms. They introduced the transformer, an encoder-decoder architecture that uses attention for connecting encoder and decoder and self-attention for generating sentence representations. Large transformer-based language models like BERT (Devlin et al., 2018) and GPT3 (Brown et al., 2020) have led to incredible progress in the field. Instead of pretraining a single layer like word embeddings, they use similar unsupervised tasks to pretrain entire deep learning models with millions of trainable parameters. They present semantic and syntactic fluency, have basic world knowledge and can be adapted to various tasks. To some extent, GPT3 can perform zero-, one-, and few-shot learning, just by intelligently formulating the question or feeding it examples as a context vector. In other words, it can solve a task without ever being trained for it by giving it one or a few examples.

By building on top of a language model, which already knows the general structure of language, I hypothesise that the disparity to the complexity of domain-specific regulations gets small enough to teach the model how to computerise building regulations with a practical amount of resources.

To the best of my knowledge, there are no literature reviews that focus on building regulation interpretation using NLP. The closest identified reviews were about representation formats for building regulations and how to use those representations for automated compliance checking. To fill this gap, I conduct a systematic literature review to identify how NLP can support or automate the interpretation of building regulations. Not only studies that cover the entire conversion process but also those that focus on a particular sub-task are of interest. Splitting the procedure into separate tasks simplifies the process and gives the opportunity to develop NLP expert systems instead of an end-to-end solution.

The following report is structured in three main parts. First, I describe the methodology used for the systematic literature review, second, the identified studies are presented and analysed, and finally, the gaps are discussed, and future research directions are suggested.

Chapter 2

Methodology

The methodology for this systematic literature review (SLR) follows in large parts the SLR guidelines in (Kitchenham, 2004). The entire process is split into four parts: 1) Preparation, 2) Literature retrieval and selection, 3) Literature analysis, and 4) Documentation. Step 2 and 3 were performed iteratively for the database search, backwards snowballing, and author search strategies.

2.1 Preparation

An initial review of the literature was conducted to define the scope of the review and to formulate the research questions. After defining search strategies, the database search was prepared by selecting a wide range of databases and empirically determining a set of keywords.

2.1.1 Research question

The focus of this review is the semantic interpretation of regulation clauses rather than structural parsing of regulatory documents. A large proportion of the initial review's papers were only partially relevant. Some studies focused on automatic compliance checking or the representation of regulations, others interpreted regulations outside of the Architecture, Engineering & Construction (AEC) domain, and only eight of the articles used NLP for the semantic interpretation of building regulations. These articles revealed that the procedure to convert building regulations into a digital representation can be split into multiple tasks. For example, J. Zhang & El-Gohary (2017) extracted information elements from building codes, transformed the elements into logic rules and aligned the logic rules to logic facts, which they extracted from a building model. Other authors classified (D. M. Salama & El-Gohary, 2016) or analysed (J. Song et al., 2018; R. Zhang & El-Gohary, 2018) building regulations to support the conversion process. Accordingly, the following research questions were formulated:

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 regulations?

2.1.2 Database selection

The topic is of a highly interdisciplinary character. It is situated at the intersection between computer science, construction, and law. A broad selection of databases helps to cover all these disciplines. They are chosen from highly construction domain-specific databases, databases covering multiple areas like science and engineering, and interdisciplinary search engines.

- American Society of Civil Engineers (ASCE) Library: Construction domain focus; 32 journals; Journal of Computing in Civil Engineering
- Engineering Village: Construction domain focus; Compendex (3,615 journals) and Inspec (nearly 5,000 journals) databases
- Scopus: Covers science and engineering; 38,589 journals
- SpringerLink: Covers science and engineering; Includes book chapters; more than 2,900 journals and 300,000 books
- ProQuest: Wide scope; 65 databases
- Google Scholar: Wide scope

2.1.3 Search terms

The default search strategy was to search only in abstract and title and use a standard set of search terms across all databases and academic search engines. However, the choice of the databases had implications on the keyword selection. For example, since SpringerLink does not offer an option for searching in abstract and title, a full-text search was used in this database. Accordingly, keywords had to be evaluated more carefully to balance the quantity and quality of search results. They had to be expressive enough to avoid an exploding number of results. Two main concepts specify the topic. Natural language processing is the overarching technology, and building regulations are the targeted text type. A sub-set of the regulation search terms divide the "building regulation" concept into the two sub-concepts, "construction domain" and "regulatory document". A tool that automated queries to SpringerLink and Scopus assisted with determining a suitable set of search terms by efficiently observing the relevance of search results based

on changes in the keywords. Good results could be achieved by using a wide range of synonyms for building regulations while avoiding the standalone use of ambiguous terms like "building", "construction", "codes", and "standards". Since the area of natural language processing is already extensive, I decided against the use of even broader terms like "machine learning" and "deep learning". Instead, the sub-tasks determined to that point in time were included (e.g. information extraction and text classification). Table 2.1 shows the final selection of search terms. The plural of each building regulation term was added, and small adjustments were made based on the limitations and functionalities of the databases. For example, controlled vocabulary was added in databases that support this feature. The appendix contains the resulting queries, adjustments, limits, and scopes per database.

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

TABLE 2.1: Search query: "NLP terms" AND ("Building regulation terms" OR ("AEC industry terms" AND "Regulation terms")

2.1.4 Criteria definition

Finally, inclusion and exclusion criteria were formulated to allow an objective selection of the literature.

Inclusion Criteria

I1) The research paper describes an approach to **automatically transform building regulations into a computable format.** This criterion aims for papers that cover the entire conversion process. These studies show how NLP can be used for the conversion (i.e. Research Question 1) and indicate the ability to automate the process completely (i.e. Research Question 3).

I2) The research paper describes a **sub- or support-task of the regulation transformation** (e.g. classification of regulations, extraction of semantic information). Compared to I1, the article does not need to have a fully digitalised regulation as an outcome. Any application of NLP to building regulations and the closely related construction contracts and utility regulations is relevant if it is likely to support the transformation.

I3) The research paper **compares NLP-based approaches to interpret building regulations**. This criterion allows the inclusion of review papers about NLP usage for building regulation interpretation. The phrase "NLP-based" enforces a focus on NLP approaches rather than NLP being one of many options for the interpretation and representation of building regulations (e.g. BuildingSMART (2017)).

Exclusion Criteria

E1) The research paper was **published before 2000**. The initial search results and the statement in Brüninghaus & Ashley (2001) that before 2001 NLP was considered not suitable to capture the complexity of legal texts indicate the suitability of this restriction.

E2) The research paper is **not available in English language**. To keep the effort manageable, the assumption that substantial research will be published in English was made.

E3) The search result is **not a journal article, conference article or book chapter** (e.g. thesis, patent, report). The assumption was made that important aspects of a thesis or a report will be published in a conference or journal. This criterion enforces a certain level of quality for the included literature.

E4) The research paper is **not related to automated compliance checking or the transformation of legal** text into a computable format. The ambiguity of certain search terms brings the need to be able to exclude numerous off-topic studies.

E5) The research paper is **about NLP and construction documents, but not about the transformation of legal text into a computable format**. This exclusion criterion includes many closely related articles. The following clusters of articles were associated with this criterion and accordingly excluded.

- Non regulatory construction documents (e.g. bridge reports (Liu & El-Gohary, 2016), accident cases (Kim & Chi, 2019), litigation cases (Mahfouz et al., 2018)) even if standards or codes are part of such documents (L. Zhang & El-Gohary, 2016a; Giuda et al., 2020)
- Document management (Cerovsek et al., 2006; Mastrodonato et al., 2010; Moon et al., 2018), document retrieval (Liang & Garrett, 2000; Lv & El-Gohary, 2016), document classification (Al Qady & Kandil, 2015), and similarity calculations and data mining on a document-level (Roshnavand et al., 2019)
- Retrieval or classification of regulation clauses for other use cases than regulation computerisation or automated compliance checking (e.g. poisonous contract clauses (Youssef et al., 2018; Lee et al., 2019))

- The creation of knowledge bases (e.g. ontologies (L. Zhang & Issa, 2011; El-Gohary & El-Diraby, 2010; T. E. El-Diraby, 2013; McGibbney & Kumar, 2015; Mahdavi & Taheri, 2018), taxonomies (T. A. El-Diraby et al., 2005; Niu et al., 2015), deontologies (D. A. Salama & El-Gohary, 2013), and epistemologies (L. Zhang & El-Gohary, 2016b)). These studies were mainly manual and comprehensive coverage cannot be guaranteed in this review. Z. Zhou et al. (2016) provides a good summary of ontology development in the construction domain.
- Semantic matching of knowledge bases for other use cases than regulation computerisation or automated compliance checking (Lima et al., 2006)
- Keyword extraction for other use cases than regulation computerisation or automated compliance checking (e.g. hazard detection in images (Tang & Golparvar-Fard, 2017))
- Regulation authoring systems (Agnoloni & Tiscornia, 2010; McGibbney & Kumar, 2013)

Studies in those areas were considered to have no direct benefit for the semantic conversion of regulatory documents and were excluded accordingly. The cited articles are only examples and do not represent the entirety of excluded articles for each area.

E6) The transformed **legal text is out of domain**. Many articles were about the interpretation of general legal texts, software development requirements, business process regulations, and more. Including all types of regulations or legal texts would go beyond a manageable scope. Furthermore, some literature and tool reviews about NLP in the legal domain (e.g. Chalkidis & Kampas (2019); MIREL (2017)) are already available.

E7) The research paper is about **automatic compliance checking** but does **not focus on the transformation of regulations**. Many of those frameworks used a hard-coded set of rules or converted the regulations manually. If no NLP usage was described in detail, the study does not provide any relevant information for this review.

E8) The research paper is about the **manual transformation or digital representation of regulations**. Although the representation formats are of high importance for the conversion, these studies do not contribute to this review when there was no NLP involved. This criterion includes the transformation of manually annotated rules to an intermediate format (e.g. RASE (Hjelseth, 2012)).

2.2 Literature retrieval and selection

Three different search strategies helped to determine the primary studies of this review. This decision was based on the complexity of choosing a suitable set of search terms.

First, most studies were identified in an extensive database search. Second, the citations in the background sections of the included papers were evaluated as potentially relevant work. Third, a literature search for authors with at least three included articles should complement the results.

2.2.1 Database search

The major decisions for the database search were made in the preparation phase of the review. The search results were retrieved on the 27th and 28th of April 2020. Figure 2.1 shows a detailed summary of the total records per databases and how many articles were excluded in each step of the exclusion procedure. The duplicate removal and data cleaning

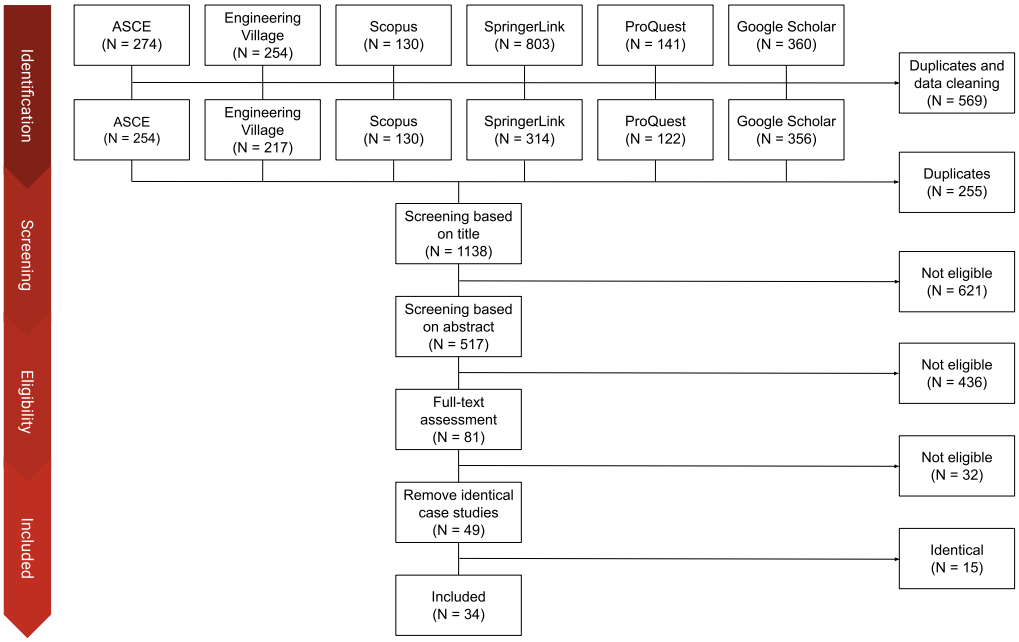

FIGURE 2.1: Flowchart of the database search process

were supported by functionalities provided by the searched databases and Mendeley, the reference manager used for this review. This step included the application of exclusion criteria E1 and E3 and setting discipline limits to Computer Science and Engineering in SpringerLink. Automatic and manual duplicate removal also accompanied the merging process of the separate database search records. In the first screening round, the literature titles were evaluated, and obvious mismatches were removed. These exclusions correspond predominantly to the exclusion criteria E2-E4. In the next step, I analysed the abstracts and excluded all papers that were clearly out of scope. Then, the full texts of the remaining articles were assessed to determine their relevance. After this step, all of the articles fulfilled the inclusion criteria I1 or I2. Since none of the candidate literature reviews focused on NLP, there is no included paper for criterion I3. Some of the articles reported

the same case study in different detail levels. For example, some authors published a conference article first and a journal article later. In that case, the less detailed or older version was excluded (i.e. the conference article in most cases).

2.2.2 Backwards snowballing

A backward snowballing strategy ensures that none of the frequently cited articles in the field is missed. Therefore, the literature identified by other authors during their background research was evaluated. This step was performed iteratively after the first full-text analysis of all included articles determined by a search strategy. Based on the papers from the database search, 51 articles were potentially relevant. Two of these articles were included in the review (i.e. L1 and L18 in Table 3.1). The others were already included, identical to an included article, or did not fit the inclusion criteria. No further items were included through backward snowballing in subsequent iterations. Since only two papers were added in this step, a good coverage by the initial database search can be concluded.

2.2.3 Author search

Searching directly for authors allows finding further research that was missed by the database search and backwards snowballing. To keep the effort manageable, I decided to use a threshold of three publication. Although this limitation might exclude individual authors, further literature from the prevalent research groups can be identified. Missing literature was searched for the authors Nora El-Gohary, Ruichuan Zhang, Zhou Peng, Jiansong Zhang, Hubo Cai, Gloria Lau and Kincho Law by using the author's publication lists, google scholar, or research gate. This search was conducted on 12 August 2020. The author search for Gloria Lau and Kincho Law identified eleven relevant articles. Nevertheless, these articles did not provide sufficient additional insights to be included compared to Table 3.1 L1, L3, and L4. Five new papers of the remaining authors were included (i.e. Table 3.1 L16, L17, L32, L34, L41), concluding to a final count of 41 articles in the review. Four of the five papers included in this step were published after the initial database search was conducted.

2.2.4 Exclusion procedure

Along with all search strategies, E1-E4 was primarily used for the screening and E5-E8 for the detailed evaluation of the literature. Since the exclusion criterion E5 was refined during the exclusion process, all studies in this category were double-checked for consistency at the end of the exclusion process.

2.3 Literature analysis process

The literature analysis was performed in three steps. First, the articles were summarised, keywords were assigned, and notes about the unique characteristics of the studies were taken. A short version of the summaries can be found in Table 3.1. The second step was to assemble the keywords to form clusters. These clusters reflect the structure of the literature and help to prepare the data extraction templates, which are partly included in this report (i.e. Table 3.1, Table 3.2, and Table 4.1). The first cluster offers a general, abstract view of the studies. It bundles the context of the studies and high-level attributes like the performed NLP tasks. This detail level is suitable to target Research Question 1. The second cluster has a technology focus and contains keywords about the rule- and machine learning-based approaches, the semantic and syntactic features, extracted information types, and the format of the transformed regulations. The last cluster contains result-focused characteristics like evaluation metrics, data sets, error sources, and limitations. It provides a view on the literature suitable to identify the gaps presented in Chapter 4. Table 2.2 provides an overview of the final categories used in the third step to systematically extract the information from the studies.

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
	Other output types	Limitations
	Used features	Contributions to the field
	Domain knowledge	

Table 2.2: Information extraction categories

2.4 Documentation

Finally, the entire review process and results are described and discussed in this systematic review report. In Chapter 3, the literature analysis outputs are arranged to target Research Questions 1 and 2. Chapter 4 will discuss the outcomes and limitations of the studies according to Research Question 2 and 3 and suggest future research directions.

Chapter 3

Literature analysis

This chapter aims to answer Research Question 1 and 2. For Question 1, a high-level overview is provided to introduce the general potential of NLP in the area of construction regulation transformation. The studies commonly split the process into several sub-tasks to allow using different NLP techniques for each task. Subsequently, a deep dive into these techniques will be utilised to discuss Research Question 2. Table 3.1 provides an overview of all included studies with their main contributions and characteristics. Furthermore, the paper-ids introduced in this table will be used throughout the results and discussion chapters. The arrangement of the entries groups the NLP tasks together and indicates their position in the conversion process pipeline.

Figure 3.1 shows that the main interest in using NLP for the computerisation of building regulations began in 2010. Before that, the regulations were usually transformed manually, and the focus of the research was to find effective digital regulation representations. This interest goes back to 1966, where Fenves (1966) used decision tables to encode design requirements. Earlier adaption of NLP to building regulations was often on a document level, where data mining techniques helped with the retrieval of clauses or documents based on similarities. For example, L1 and L3 parsed regulation documents and created a feature enriched XML-repository with compliance assistance being one of the possible use cases. The features were then used to compare clauses from various regulatory documents. The research interest reached a peak in 2016 and has remained high since then. This development could be explained by the overall progress in NLP, the maturity of free NLP tools, and deep learning being successfully applied for NLP.

TABLE 3.1: Overview of included studies. Ordered by NLP task, year and author. *: Task is covered by a newer study and excluded from counts in Figure 3.2. Characteristics: 1 - International Building Codes, 2 - Energy codes and standards, 3 - Construction contracts, 4 - Accessibility regulations, 5 - Utility regulations, 6 - Korean building act, 7 – Others; a - Automated Compliance Checking, b - Compliance checking of underground infrastructure, c - Project scope comprehension, d - E-Rulemaking, e - Automated construction specification review, f - Construction quality management system, g - Model checking, h - Compliance assistance, i - Regulation retrieval, j - Knowledge graph; I – semi-automated, II – ontology-based, III – NLP tools and rules, IV – Machine learning, V – Deep learning, VI – similarity-based

ID	NLP Tasks	Comprehension	Characteristics	Validation Results	Reference
L1	Document processing Feature extraction Similarity analysis*	They use a shallow parser to convert text documents into an XML structure and enrich the regulation repository semi-automatically with features like term definitions, references, measurements, and concepts. The features are then used to calculate the similarity between provisions.	4, d, h, i, I, III, VI	-	(Lau & Law, 2004)
L2	Preprocessing	Word segmentation of Chinese building design specifications to support future named entity recognition for knowledge graph development. They remove non-Chinese characters and segment the resulting character sequence by matching sequences to a hashed dictionary. They start with a character sequence of maximum length and shorten it until getting a match.	7, j, III	Experiments on max word length and running time	(J. Zhang et al., 2018)
L3	Similarity analysis	Extension of the similarity analysis of L1. They calculate the similarity between provisions based on general, domain, and structural features. This similarity score was then refined based on neighbour provisions and references.	4, d, i, VI	Root mean square error (RMSE): 22.9	(Lau et al., 2006)
L4	Similarity analysis	Framework for taxonomy-based building regulation retrieval. They use a simple keyword latching process to match one regulation with one taxonomy, a regulation clustering based on Lau et al. (2006) and pivoting from the most familiar regulation for 1-n concept-section mapping, and taxonomy mapping for n-1 concept-section mapping.	1, i, VI	Ontology mapping: RMSE: 9.8 F-measure: 73%	(Cheng et al., 2008)
L5	Similarity analysis Information alignment	Semantic analysis using a word2vec model to support the transformation of regulations. A target word can be entered, and the system shows the most related noun phrases, similar sentences in the Korean building act, and associated concepts in the KBim object/property database.	6, a, VI	-	(J. Y. Song et al., 2018)
L6	Similarity analysis* Text classification	Extends L5 with multi-class classification for topic prediction. They built a deep learning classifier consisting of four fully-connected layers and a softmax layer.	6, a, V, VI	-	(J. Song et al., 2018)
L7	Similarity analysis	They clustered building code requirements into seven categories and annotated the clauses with semantic information elements and RASE. Depending on the required time and ability to annotate the clauses, they determined the computability of the clusters.	1, a, IV	Silhouette coefficient: 96.2%	(R. Zhang & El-Gohary, 2018)
L8	Text classification	They compared different ML-algorithms and configurations for the binary classification of clauses from general conditions of construction contracts into environmental and non-environmental. The best configuration used a support vector machine (SVM) with bag-of-words (BOW) representations and the 20 best features.	3, a, IV	Precision: 96% Recall: 100%	(D. M. Salama & El-Gohary, 2016)

Table 3.1 continued from previous page

L9	Text classification	The method is based on L8, but instead of multiple binary classification tasks they perform a multi-label classification of clauses into ten environmental topics. Their experiments confirm the selection of SVMs and BOW.	1, a, IV	Precision: 84% Recall: 97%	(P. Zhou & El-Gohary, 2016a)
L10	Text classification	Ontology-based multi-label classification into six environmental topics. They calculate sentence representations with a skip-gram model and calculate similarities between the sentence representations and ontology concepts. They were able to slightly outperform an ML-based classifier (L9).	1, a, II, V, VI	Macro-based metric: Precision: 90.4% Recall: 97.7%	(P. Zhou & El-Gohary, 2016b)
L11	Text classification	Binary classification of contract clauses into requirements and non-requirements. They conducted experiments on feature selection for Naïve Bayes and chose uni-grams over bi- and tri-grams.	3, c, IV	Accuracy: 91.5%	(Le et al., 2019)
L12	Text classification	Performance evaluation of manual, rule-based and different ML-based classification approaches for the classification task in L11. SVM with uni-grams and lemmatisation achieved the best results.	3, c, IV	Accuracy: 98.2%	(Hassan & Le, 2020)
L13	Feature extraction Information extraction*	Comparison between phrase structure grammar (PSG) and dependency grammar (DG) as features for rule-based information extraction. While the Stanford dependency parser made fewer parsing errors, the rule-based generation of PSG allowed more flexibility.	1, a, III	F-measure PSG: 94.3% F-measure DG: 96.9%	(J. Zhang & El-Gohary, 2012)
L14	Feature extraction	Semantic role labelling of building codes to support information extraction. Out-of-domain training data was pruned to improve performance. They used conditional random fields (CRF) and a wide range of features like POS tags, phrase tags, and dependency trees.	1, a, IV	Precision: 71% Recall: 63%	(R. Zhang & El-Gohary, 2019a)
L15	Feature extraction	Unsupervised machine learning to generate templates that can enhance the information extraction process. Sentence fragments from IBC were clustered based on semantic role labels. The resulting eight clusters were turned into templates by identifying patterns for fixed parts and semantic information elements.	1, a, IV, VI	Accuracy: 76%	(R. Zhang & El-Gohary, 2019d)
L16	Feature extraction	Accuracy comparison of seven POS taggers on tagging building codes. The gold standard was developed by automatically annotating building codes with all machine taggers and manually resolving disparities. The Stanford CoreNLP tagger achieved the highest accuracy. The main error sources were word ambiguities, rare words, and unique word meanings.	1, a, III	Accuracies: Average: 88.8% Best: 89.8% Combined: 90.2%	(Xue & Zhang, 2020b)
L17	Feature extraction	Rule-based error correction algorithm to improve the quality of POS tags for building codes. Their algorithm generated 14 rule set templates, resulting in 895 rules that can be used to fix incorrect POS tags.	1, a, III	Accuracy: 89.4% -> 98.1%	(Xue & Zhang, 2020a)

Table 3.1 continued from previous page

L18	Information extraction	Extraction of concept relation triplets from construction contracts using a shallow parser (i.e. Sundance (Riloff & Phillips, 2004)). The parser divides sentences into clauses, identifies phrases, and analyses the phrases' roles, part-of-speech of words, and more. The phrases and roles were used in an algorithm to determine the active concept, the passive concept, and the relation.	3, i, III	Precision: 70% Recall: 67%	(Al Qady & Kandil, 2010)
L19	Information extraction	Rule- and ontology-based extraction of five information types. They compared the adaptability of an ontology-based method with their previous approach, which used domain gazetteers. Therefore, they used different types of regulatory documents for development and test.	7, a, II, III	Precision: 92.9% Recall: 86.7%	(Kwon et al., 2013)
L20	Information extraction Information transformation	Semi-automatic approach to transforming natural language constraints into trees. The mapping information is stored in a database. Unknown information elements need to be added to the database by a user.	7, g, I, III	Transformation: 53 out of 83 constraints	(Niemeijer et al., 2014)
L21	Information extraction Information transformation	NLP tools, rules, and ontology-based mapping to extract information from regulations. The elements were then transformed to RAINS, a controlled natural language that allows automatic conversion to SPARQL.	7, a, II, III	-	Emani et al. (2016)
L22	Information extraction	Ontology-based extraction of information to support experts in transforming regulations into SWRL or MVDXML rules. The transformed rules were then used as input for MVDXML checker and a custom SWRL rule engine to execute the rules.	7, a, I, II	-	(Fahad et al., 2016)
L23	Information extraction Information transformation	Syntactic features and gazetteer lists were used for the rule-based information extraction and transformation. They represented the spatial rules in ten information tuples (e.g. hierarchy, subject, landmark, spatial relation). This representation was then used for spatial reasoning in GIS.	5, b, III	Transformation: Precision: 87.9% Recall: 79.1%	(S. Li et al., 2016)
L24	Information extraction	Semi-automated framework for knowledge-driven building design checks. They identify the intended meaning of terms using WordNet, NLTK, and ontology matching. The Stanford CoreNLP dependency parser and custom models pre-annotated the analysis procedures (i.e. expert knowledge) with input, output, applicability, selection, and negation. Performance indicators (e.g. building codes) were then pre-annotated with subject, comparator, and object.	7, g, I, II, III	-	(Mathot et al., 2016)
L25	Information extraction	The semantic information elements (i.e. subject, subject restriction, compliance checking attribute, deontic operator indicator, quantitative relation, comparative relation, quantity value, quantity unit/reference, and quantity restriction) defined in this study were used by numerous studies subsequently. Furthermore, they introduced a rule- and ontology-based information extraction method. Overall, they used 146 extraction patterns, 187 features, and 38 conflict resolution rules.	1, a, II, III	Precision: 96.9% Recall: 94.4%	(J. Zhang & El-Gohary, 2016b)

Table 3.1 continued from previous page

L26	Information extraction Information alignment	Ontology-based information extraction supports experts with the transformation of regulations into SWRL. Additionally, a mapping between ontology concepts and the Industry Foundation Classes (IFC)-schema facilitates the manual transformation from SWRL to executable rules in XSLT.	4, a, I, II	-	(Shi & Roman, 2017)
L27	Information extraction Preprocessing	Based on the rule-based information extraction of L25. They leverage domain-specific preprocessing, ontology-based matching, and sequential and cascaded extraction to handle the higher complexity of energy codes and to preserve the inherent hierarchy of regulations.	2, a, II, III	Precision: 98.5% Recall: 97.4%	(P. Zhou & El-Gohary, 2017)
L28	Information extraction Similarity analysis	The paper compares specifications on a document, requirement and entity level. For the entity level comparison, they used a recurrent neural network for named entity recognition of six categories.	3, e, V, VI	Precision: 21.1% Recall: 35.7%	(Moon et al., 2019)
L29	Information extraction Feature extraction	Frames and gazetteer lists for the extraction of spatial information. They identified frames like "within distance" and "deontic rules", which can be invoked by looking up a target word. The frame slots are then filled using syntactic features and semantic role labelling. The evaluation of the approach was limited to the information extraction using the "within distance"-frame.	5, b, II, III, IV	Precision: 92.3%	(Xu & Cai, 2019)
L30	Information extraction Feature extraction Information transformation	They transformed the regulation clauses to context-free grammar to allow hierarchical extraction of spatial information elements. A set of rules was used to formalise the extracted information in deontic logic. This format allows automatic conversion into database triggers using existent tools.	5, b, III	Precision: 95.3% Recall: 74.2%	(Xu et al., 2019)
L31	Information extraction Feature extraction	Development of a deep learning dependency parser using pruned out-of-domain training data. They developed an algorithm that uses the dependency trees to segment the regulation clauses into requirement units and to interpret the relationship between the units. Furthermore, they showed that their custom dependency parser outperforms the Stanford dependency parser when applied to this extraction task.	1, a, III, V	Average normalised edit distance: 0.32 Precision: 89% Recall: 76%	(R. Zhang & El-Gohary, 2019c)
L32	Information extraction	Transfer learning and deep learning to annotate building codes with semantic information elements. They used a Bi-LSTM-CRF architecture and 20.000 POS tag annotated sentences from Penn Treebank, and conducted experiments with different transfer learning strategies.	1, a, V	Precision: 88% Recall: 86%	(R. Zhang & El-Gohary, 2020b)
L33	Information extraction	Extraction of procedural constraints from construction regulations using a Bi-LSTM-CRF model for named entity recognition. The relationship between these constraints was determined using a Bi-LSTM-MLP classification model.	7, f, V	Entities and relations: Precision: 73.9% Recall: 73.9%	(Zhong et al., 2020)

Table 3.1 continued from previous page

L34	Information transformation*	Comparison of a bottom-up and a top-down approach for the information extraction and transformation. In the top-down approach, the entire restrictions were extracted and split during the transformation. In the bottom-up approach, information about relationships and syntactic features were extracted and used to transform the restrictions.	1, a, III	F-measure top-down: 93.9% F-measure bottom-up: 96.2%	(J. Zhang & El-Gohary, 2013)
L35	Information transformation	Rule-based transformation of semantic information elements (L25) into Prolog logic rules using a bottom-up strategy (L34). They defined 40 semantic and syntactic information tags, rules to resolve conflicting tags, and semantic mapping rules that utilise the tags for the conversion.	1, a, III	Precision: 98.2% Recall: 99.1%	(J. Zhang & El-Gohary, 2015)
L36	Information alignment	Matches regulation keywords to building concepts to objectively suggest the IFC-schema extensions that are necessary for automated compliance checking. Therefore, they extracted concepts from regulations and IFC, conducted term-based and similarity-based matching, and classified the relationships with a machine learning algorithm.	1, a, IV, VI	IFC concept selection - adoption rate: 84.5%	(J. Zhang & El-Gohary, 2016a)
L37	Information alignment	Matching of regulation information elements and IFC concepts and relations using the buildingSMART Data Dictionary (bSDD), ontologies, and Word2Vec embeddings. They identified candidate concepts using three strategies to lookup bSDD and ontologies and calculated the similarity scores to select the best match.	2, a, II, VI	Precision: 98.0% Recall: 89.2%	(P. Zhou & El-Gohary, 2018a)
L38	Information alignment	Domain and general word embeddings, cosine similarity, and supervised machine learning were used to match regulation concepts and relations with their IFC equivalents.	1, a, IV, VI	Accuracy: 77.5%	(R. Zhang & El-Gohary, 2019b)
L39	Information extraction* Information transformation* Information alignment	Integration of their previous efforts for information extraction (L25) and transformation (L35) with the extraction of logic facts from IFC files into an automated compliance checking framework. They conducted an end-to-end validation of the framework to identify non-compliant instances in a building model.	1, a, II, III	End-to-end: Precision: 87.6% Recall: 98.7%	(J. Zhang & El-Gohary, 2017)
L40	Text classification* Information extraction* Information alignment* Information transformation	Integration of text classification (L10), information extraction and transformation from regulations (L27) and IFC, and information alignment (L37) into a unified automated compliance checking system and end-to-end evaluation of the framework.	2, a, II, III	End-to-end: Precision: 84.6% Recall: 91.7%	(P. Zhou & El-Gohary, 2018b)
L41	Quality Assurance	Natural language generation to recreate building code sentences from semantic information elements. ROGUE and BLEU scores are used to evaluate the generated sentence and with it, the comprehensiveness of the extracted information.	1, a, VI	ROUGE1: 86% ROUGE2: 78% BLEU1: 80% BLEU2: 73%	(R. Zhang & El-Gohary, 2020a)

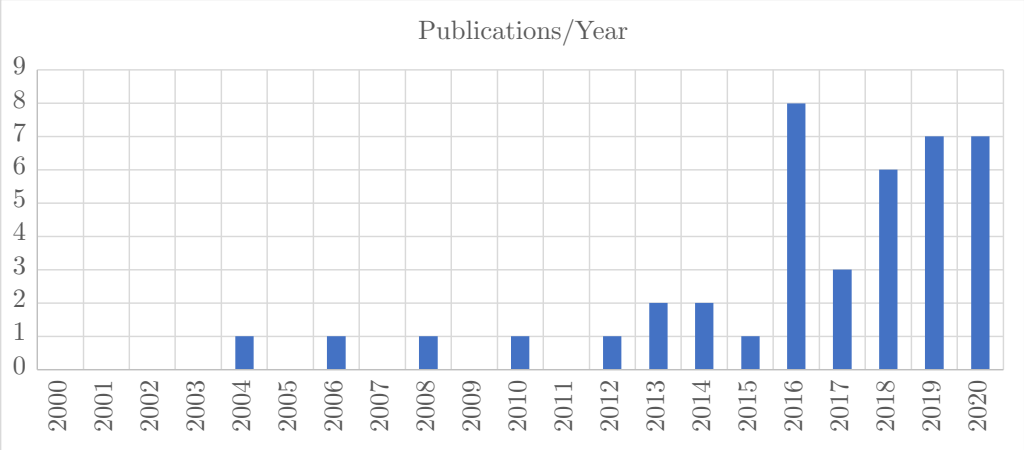

FIGURE 3.1: Number of included publications per year since 2000

Also, the level of automation changed over time. In the early 2000s, the trend in the included papers aimed at compliance assistance. Later the intention was to create fully automated compliance checking systems. Some approaches support experts in creating digital codes by extracting concepts and properties from regulations (L5, L6, L22, L26). In addition, L5 and L6 offered further context information like related regulations and a topic classification. The semi-automated approaches L20 and L24 only required user-input for missing phrases in the knowledge base and L36 to review the NLP output. For the fully automated methods, selecting a technology determines the source and amount of development work required. Those expenditures include developing rules, capturing domain knowledge with ontologies, selecting features for ML-algorithms, labelling training data for supervised learning, developing deep learning architectures, and optimising hyper-parameters.

Further distinctions can be drawn in the context of the studies and the used legal documents. While most of the studies were about compliance checking of buildings, there were some exceptions to this use case. For example, L23, L29, and L30 applied their approach in the domain of utility compliance checking. Utility regulations mostly contain spatial constraints between utilities and landmarks, and utility design information typically comes from a Geographic Information System (GIS). Other studies utilised requirement classification (L11, L12) and information extraction (L28) for systems to comprehend or review construction contracts. Moreover, L33 extracts construction procedures for construction quality management. Here, the focus lies in the temporal constraints between those procedures.

The building compliance checking approaches also show variations in the type of legal document they used. On the one hand, there are different types of legal documents, like building acts, building codes, standards of acceptable solutions, and norms. Different types of normative texts potentially vary in structure, vocabulary and complexity. On the other hand, the research conducted in this area is spread all over the world. Accordingly, authors usually use regulations that are in their language and applicable in their country.

The articles about end-to-end compliance checking frameworks and case series of authors working towards such frameworks offer an excellent overview of the tasks that are frequently supported by, or performed with, NLP. With three case series about NLP to convert building codes, major research interest is situated in the United States. Jiansong Zhang and Nora El-Gohary built a rule-based framework for automated compliance

checking. They reported their efforts in papers about information extraction (L13, L25), information transformation (L34, L35), the full compliance checking framework (L39) and a proposal for objective extension of the IFC-scheme by aligning regulations concepts with IFC concepts (L36). Peng Zhou and El-Gohary built on top of this approach to deal with the higher complexity of environmental regulations. They started with the classification of regulations to filter out irrelevant clauses (L9, L10). For the information extraction, they shifted the focus to using ontologies and the preservation of the hierarchical structure of the regulation clauses (L27). Furthermore, they explored a sophisticated information alignment approach in L37 to bring the building and regulatory information together before combining all the steps to the final framework in L40. The most recent case series of Ruichuan Zhang and Nora El-Gohary shifted the attention towards deep learning technologies to convert the building regulations. They started with structural (L7) and semantic (L14, L15) analysis of the regulations and extracted the hierarchies (L31) and semantic information elements (L32) of the requirements in separate information extraction steps. Then, they assessed the quality of these semantic information elements by recreating the original sentences using natural language generation before finally aligning the information streams in L38. Other groups targeted the full automated compliance checking process in their articles, primarily focusing on information extraction, transformation and alignment (L22, L23, L26, L30). L22 and L23 also introduced methods for the reasoning task, but an end-to-end validation is missing. Figure 3.2 brings all sub-tasks into a sequence giving an overview of the entire automated compliance checking process. This includes the reasoning step and the information extraction and transformation from a BIM. The number of included articles, which report about an NLP task in detail, is presented in brackets. These counts should be taken with care, considering the focus of this review is the semantic parsing of building regulations. Accordingly, it is likely that fewer studies about document processing were detected by the selected search terms. In addition, since the building design information is not considered to be in natural language, studies focusing on the information extraction and transformation from IFC are excluded.

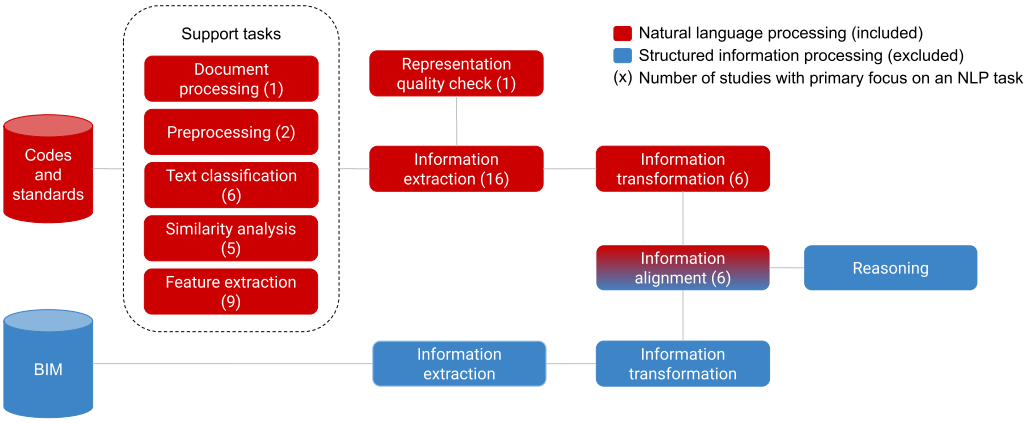

FIGURE 3.2: NLP supported automated code compliance checking process.

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. One of the preconditions for the computerisation of codes is the existence of the documents in a structured textual form. Many of those are only available in PDF- or HTML-format. Most NLP-based computerisation approaches like L39 and L40 used a set of regulation clauses to create their ground truth. They collected these clauses either manually (L10) or with a simple algorithm applied to regulatory documents (L8, L40).

L1 was the only study to focus on this task. They developed a parser to transform regulations from HTML, PDF, or plain text into an XML format. The XML structure preserves the inherent hierarchy of the regulations and is augmented with additional features like references, concepts, and exceptions. There are undoubtedly further studies that focus on the structural parsing of regulation documents. Still, since the search keywords were directed towards semantic parsing of clauses, and studies before 2000 and with the focus on information retrieval outside of the compliance checking context were excluded, this remains the only example.

3.2 Preprocessing

Preprocessing prepares the input text for NLP models or algorithms. Stanford CoreNLP (Manning et al., 2014), GATE tools (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. Usually, there are more preprocessing steps in rule-based and machine learning-based methods than in deep learning since deep learning models benefit from a higher information level. As machine learning algorithms often perform better with fewer, more meaningful features, it is common to remove stop words and use word stems. The included studies used the following preprocessing steps:

- Sentence splitting: The text is split into sentences with punctuation marks indicating the sentence boundaries (e.g. L23, L25, L29, L30, L33). L27 developed a domain-specific sentence splitting that can handle regulatory cross-references, which often include a period (e.g. AS/NZS 3661.1).
- Tokenisation: The sentences are split into word tokens based on white spaces (e.g. L8, L10, L14, L22, L23, L25, L27, L28, L29, L30, L31). Tokenisers can vary for special cases like "aren't", punctuation marks, and special tokens like sentence start and uppercase words.
- Dehyphenation: Reverses the hyphenation of words (e.g. L25, L23, L29, L30).
- Lowercasing: Transforms all word tokens to lowercase. This process is often integrated with the tokenisation. The information can be preserved in special tokens if relevant to the task (e.g. L9, L12, L31, L14).
- Singularising: Transforming plural words to their singular form (e.g. L38).
- Morphological analysis: Transforming words into their basic form (e.g. L6, L25, L29, L30, L37). Stemming is the rule-based version where amongst other things, suffixes are cut off (e.g. L4, L8, L10, L12, L14, L22, L28, L31, L36), and lemmatisation is the more sophisticated version, where the root of a word is identified (e.g. L12). For example, the lemma of "better" is "good".
- Special characters removal: This preprocessing step is often included in the tokenisation. For example, L10 and L12 removed punctuation marks, and L6 removed Chinese characters, punctuation marks and numbers.

- Stop word removal: High-frequency words like "be", "have", "and", "by" carry less meaning and are often removed for tasks like text classification (e.g. L6, L8, L10, L12, L22, L28).
- Rare word removal: Especially in deep learning, rare words are often removed or replaced by special tokens to keep the vocabulary manageable and avoid word vectors that do not carry enough meaning (e.g. L6, L12, L41).
- Tagging and parsing: In many cases, part-of-speech (POS) tagging (e.g. L6, L7, L12, L15, L30, L31, L36), phrase and clause tagging (L7), constituency parsing (e.g. L7, L15), and dependency parsing (e.g. L13) were considered as preprocessing. More details will be given in the sections about feature and information extraction.
- Domain-specific preprocessing: L27 removed quotation marks and parenthesis, including the information inside of the parenthesis. Furthermore, they split the regulation clauses into exceptions, conjunctions and lists and stitched the heading and relationship indicators. In contrast, L18 split the lists manually and attached the section numbers to each clause.

While the tokenisation of English text can be seen as a solved task, languages without white spaces between words are much more complex to parse. Common approaches are based on rules, statistics, or dictionaries. L2 researched the tokenisation of Chinese building specifications with a hashed dictionary and a reverse maximum matching algorithm.

3.3 Similarity analysis

A common technique for clustering and information retrieval is to evaluate the similarity of text based on word stems or vector representations. L1 and L3 used the feature-enriched XML repository described in Section 3.1 to calculate the similarity between regulation clauses. They refined the similarity scores using features like parent and sibling clauses, references, and definitions from the regulatory documents and book glossaries. L4 leveraged the relatedness analysis of L3 for taxonomy-based regulation retrieval. L5 and L6 introduce an expert support system for the regulation transformation. While transforming a regulation clause, the system provides context information like related regulations, classifies the clause by topic, and allows the user to search for concept-related nouns and predicates. Furthermore, L7 used unsupervised machine learning to cluster building codes to determine differences in complexity and structure, and consequently, in computability. They identified seven sentence types, which differed in length, the existence of independent and dependent clauses, missing essential information, and the existence of secondary information elements (i.e. restrictions). 60% of the sentences account for simple requirements with one independent clause, no secondary information elements, and high to medium computability. This information is helpful for creating data sets and determining the level of automation that can be achieved with automated compliance checking.

3.4 Feature extraction

A specific type of text analysis is the determination of semantic and syntactic constituents of sentences. Structural features like POS tags, phrase chunks, and dependency trees were commonly used for rule-based information extraction (see Section 3.6 for further details).

L13 compared the suitability of phrase structure grammar and dependency grammar for feature-based algorithms. They used the Stanford dependency parser to generate dependency trees and a custom set of rules to determine phrase chunks. Phrase structure grammar is more adaptive to a domain, but it is more error-prone since the rules are less tested than frequently used open-source tools. Overall, they achieved better results with dependency grammar (i.e. 96.9% F-measure) than with phrase structure grammar (i.e. 94.3% F-measure).

Although there are various open tools for dependency tree parsing, there are reasons to develop a custom dependency parser. On the one hand, L31 optimised the dependency parser for building codes and made use of new neural network architectures. A deep learning dependency parser was developed to create features for a rule-based requirement unit extraction. They used a general data set and pruned training samples that were not similar enough to their development set of building codes. Using the deep learning dependency parser instead of the Stanford dependency parser to generate features for the requirement unit extraction accounted for an overall improvement of 2%. On the other hand, L30 tagged the regulations with POS tags, phrasal tags and domain-specific tags. They developed a rule-based dependency parser to make use of these tags for the dependency tree generation.

Wrong POS tags were a common error source in rule-based information extraction (L25, L27). L16 evaluated seven state-of-the-art POS tagger on building codes and identified word ambiguities, rare words, and unique word meanings as the main cause for wrong POS tags. L17 aimed to resolve this issue by introducing an error correction algorithm to fix the POS tag errors that commonly appear with building codes. They were able to enhance the accuracy from 89.4% to 98.1%.

A method to create semantic features is semantic role labelling, an area with large established data sets like FrameNet (L29) and PropBank (L14). Semantic role labelling aims to label a sentence with roles like agents (i.e. the subject that is performing an action), recipients (i.e. the entity that is targeted by an action), actions, and various modifiers (e.g. adverb, location, manner, and temporal). Those roles are related to the information types that are required for compliance checking. L14 achieved an F1-measure of 65% labelling building codes. Compared to F1-measures between around 85 to 90% in (Z. Li et al., 2020) on the WSJ and Brown data sets, this score is relatively low. L15 used the semantic role labels in combination with syntactic features to identify templates that can support the information extraction task automatically. They generated eight templates achieving an accuracy of 76%. The use of these templates for information extraction is, to my knowledge, still pending.

3.5 Text classification

Text classification was a common technique to support the transformation process by filtering out irrelevant clauses. L8-L12 used the classification to filter out clauses that do not contain requirements. Furthermore, L6 and L8-L10 classified the clauses by topic. These topics are beneficial for later stages of compliance checking when a building model should only be checked for compliance with a specific set of rules. L8, L9, L11 and L12 used machine learning (ML) for the classification task. They performed a wide range of experiments to identify the best ML-algorithms, features, and feature weighting and selection methods for the task. Multiple papers (L8, L9 and L12) identified support vector machines as the best suited ML-algorithm.

In contrast, L6 and L10 avoided feature engineering by using deep learning. Since L6 did not publish any evaluation metrics, it cannot be compared to the other approaches.

L10 tried to tweak the performance by incorporation domain knowledge (i.e. ontologies). They used unsupervised deep learning to generate representations of the clauses and the ontology concepts and calculated the similarities between them. Most of the authors agreed to focus on a high recall (i.e. 100% recall was desired) since a falsely classified clause could cause the system to miss requirements and non-compliant building specifications. L8 showed that this goal is achievable for binary classification tasks (i.e. environmental or non-environmental). The highest recall for the classification into requirements and non-requirements was 95% (L12). The multi-label classification of clauses into different environmental topics like air leakage and thermal insulation could also achieve a high average recall of around 97% with the ontology-based approach in L10 slightly outperforming the machine learning-based method introduced in L9.

3.6 Information extraction

Information extraction is a generic term for the identification of semantic information elements, requirements units, events, and relations in unstructured text. Since information extraction is the common task among all regulation computerisation approaches, it received the highest research interest (L19-L33). Table 3.2 gives a detailed comparison of the technical details of those studies. The studies that perform information extraction from building regulations differ substantially in the depth of the extracted information. The differences range from extracting concepts and relations (e.g. L18, L22, L26), to the extraction of a medium level of information like building element, quantity, property, relation, and function to check simple requirements (e.g. L19, L20, L21), to an extensive information extraction of around ten different information elements from complex requirements (e.g. L25, L27, L32). Figure 3.3 shows a simple example annotated with the information elements defined in L25. L23, L29, and L30 adapted these information elements for spatial constraints prevalent in utility regulations. In contrast, L28 extracted information from construction contracts, and L31 extracted construction procedures from regulations. Thus the information elements used in those two studies are less comparable. The restrictions contained in more complex requirements were handled in various ways. Either the restrictions were extracted directly (e.g. subject and quantity restrictions in L25), they were determined as a composition of other information elements (L27), or the clauses were split into requirement units first (L31), and those units were the input for further information extraction (L32).

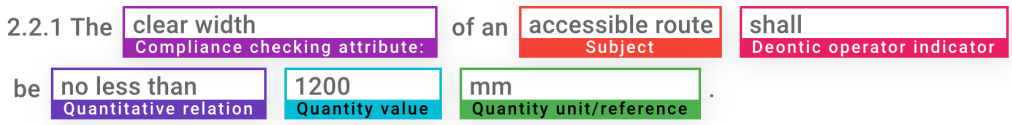

Figure 3.3: Simple regulation clause 2.2.1 from Ministry of Business, Innovation and Employment (2014) annotated with information elements from L25. Annotations made with doccano (Nakayama et al., 2018).

TABLE 3.2: Information extraction techniques. *: Assumption (i.e. no details available)

ID	Technology	Process steps	Technology stack	Extracted information	Used features	Domain knowledge
L18	NLP tools, rules	1. Input file preparation (section number, tokenisation, resolving lists) 2. Shallow parsing (clauses, phrases, phrase roles, POS tags) 3. Concept set extraction	2. Shallow parser sentence understanding and concept extraction (Sundance) 3. Extraction algorithm	Active concept, relation, passive concept	Clauses, phrases, phrase roles	
L19	NLP tools, ontology, rules	1. Preprocessing (tokenisation, sentence splitting) 2. Syntactic features (POS tags) 3. Semantic features (named entity recognition) 4. Information extraction (IE)	3/4. Handcrafted rules, GATE tools: JAPE transducer, OntoRoot Gazetteer	Building element, quantity, property, relation, function	3/4. Gazetteer, ontology 4. POS tags, named entities	Modified PROTON ontology
L20	Semi-automatic, NLP tools	1. Tokenisation 2. Word lookup	2. Database	Element, unit, value, relationship, comparison, operator (i.e. add/subtract, multiply/divide, boolean)	2. Term lists	Database
L21	NLP tools, ontology	1. Assertion identification 2. Phrase detection 3. Lookup-based IE* 4. Ontology alignment 5. Identify main predicate/concept	1. Open information extraction (CSD-IE) 2. Illinois chunker 4. Stop word removal, n-gram string similarity 5. Stanford parser	1. Assertions 3. Negation marker, comparators, literals 4. Object/datatype property, concept/individual 5. Main predicate, main concept	3. Phrase chunks, term lists* 4. Phrase chunks, ontology 5. Dependencies	IfcOWL
L22	NLP tools, ontology	1. Preprocessing (tokenisation, stopword removal, stemming) 2. Ontology alignment	1. FrenchStemmer	Individuals (Uncertain: concepts, object property, datatype property)	Ontology	IfcOWL, bSDD and SKOS ontology
L23	NLP tools, rules	1. Preprocessing (sentence segmentation, tokenisation, dehyphenation) 2. Feature creation (POS tagging, gazetteer lookup, chunking) 3. Hierarchical information extraction	1/2. NLTK 2. 38 chunking rules, regular expressions 3. Extraction rules, rule execution hierarchy	Requirement hierarchy, subject, deontic indicator, verb, spatial descriptor, landmark, quantitative value	POS tags, general gazetteer (negation, comparative relations and lists), domain gazetteer, phrases	Domain-specific terms (utility types, attributes, components, events), spatial descriptors

Table 3.2 continued from previous page

L24	NLP tools, ontology	1. Identify word meaning (stemming, word lookup, ontology matching) 2. Resolve unidentified words (manual) 3. Semantic annotation 4. Manual review of annotations	1. WordNet, NLTK 2. Ontology editor 3. Stanford CoreNLP dependency parser, custom models 4. BRAT	Analysis step: input, output, applicability, selection (comparable with restrictions), negation Performance indicator: subject, comparator, object	WordNet synsets, ontology	(Draft) standard ontologies, ontology editor
L25	NLP tools, ontology, rules	1. Preprocessing (tokenisation, sentence splitting, morphological analysis, dehyphenation) 2. Feature generation 3. Rule development 4. Information extraction 5. Evaluation	GATE tools (ANNIE, morphological analyser, ontology editor, JAPE transducer) 2. Phrase structure rules 4. Information extraction rules, conflict resolution rules, information extraction hierarchy	subject, compliance checking attribute, deontic operator indicator, quantitative relation, comparative relation, quantity value, quantity unit, quantity reference, subject restriction, quantity restriction	Gazetteer lists (comparative relation list, unit list, negation list, GATE numbers and ordinals), POS tags, phrase structure grammar pattern, ontologies	Developed in OWL; Reuses IC-PRO-Onto, IFC concepts
L26	Ontology	Ontology mapping to extract information	GATE tools: ANNIE, ontology editor	Classes, properties, individuals	Ontology	Ontology extended with Wikipedia and regulation terms
L27	NLP tools, ontology, rules	1. Domain-specific preprocessing (i.e. splitting and stitching, removal of parenthesis and quotation marks) 2. Preprocessing (tokenisation, sentence splitting, morphological analysis) 3. Feature creation 4. Sequential dependency-based extraction	Regex, GATE tools: ANNIE, JAPE, etc. 2. Sentence splitting rules 3. Tagging rules (i.e. auxiliary tags) 4. Information extraction rules, conflict resolution rules, extraction sequence, cascaded extraction	1. Provisions, exceptions, relationships 4. subject, compliance checking attribute, deontic operator indicator, quantitative relation, comparative relation, quantity value, quantity unit/reference, subject restriction, quantity restriction	POS tags, gazetteers (negation, measurement units, location, currency, etc.), 15 auxiliary tags (e.g. list number and cardinal number), ontology, extracted information	Ontology (OWL): Ninth level, 335 concepts, commercial building energy efficiency domain
L28	Deep learning	1. Preprocessing (tokenisation, stopword removal, stemming) 2. Named entity recognition	2. RNN	Object, standard, environment, condition, reference, none	Word embeddings	Labelled training data

Table 3.2 continued from previous page

L29	Machine learning, NLP tools, rules, frames	1. Preprocessing (tokenisation, sentence splitting, morphological analysis, dehyphenation) 2. Feature creation 3. Identify frame target words and elements 4. Semantic role labelling of frame elements 5. Information extraction	1. Stanford tools 2. Stanford parser 3. Rules, regular expressions 4. ML with FrameNet training data 5. Mapping rules to identify frame elements	Deontic element, lexical element, composition element, qualitative element, quantitative element, subject, checked attribute, subject restriction	3. Frames, POS tags, gazetteer lists (i.e. negations, comparators, units), phrases, ontology 4. Phrase type, parse tree path, position, voice, target word, domain concept class, domain relationship class, gazetteer class 5. Frame elements	Ontologies
L30	NLP tools, rules	1. Preprocessing (sentence segmentation, tokenisation, dehyphenation, POS tagging) 2. Gazetteer compiling 3. Syntactic parsing 4. Information extraction	3. Stanford parser vs rule-based parse tree generation, context free grammar 4. Functions	Hierarchy, spatial relation (topology, location) trajectory, spatial indicator, landmark, distance value, comparative relation, unit	3/4. POS tags, gazetteer lists, phrasal tags 4. Parse trees	Gazetteer (domain physical products, spatial prepositions, spatial verbs, comparative relations, negations, units)
L31	Deep learning, rules	1. Data preparation and preprocessing 2. Data adaptation 3. Deep learning dependency parser 4. Requirement segmentation 5. Restriction interpretation	1. Out-of-domain training data (English Treebank of Universal Dependencies) 2. Similarity-based data pruning, word2vec 3. Embedding, LSTM, MLP, hyperparameter optimisation 4/5. Rules	4. Requirement units 5. Relationship (i.e. limitted to restrictions in this study)	3. Word and POS tag embeddings 4/5. Dependency tree	2. Domain training data
L32	Deep learning	1. Data preparation 2. Model development 3. Model training 4. Information extraction	2. Input embeddings, Bi-LSTM, CRF, Keras, Python3, TensorFlow 3. Transfer learning, two-stage training vs alternating training	Subject, compliance checking attribute, deontic operator indicator, quantitative relation, comparative relation, quantity value, quantity unit, subject relation, syntactic fillers	Penn Treebank dataset annotated with POS tags	Annotated building code sentence fragments
L33	Deep learning	1. Data preparation 2. Model training 3. Named entity recognition 4. Classification	2. Pytorch 3. Bi-LSTM-CRF 4. LSTM-MLP	3. Construction procedures, construction objects, interval time, others 4. Relationship	3. 50-dimensional character embeddings 4. 50-dimensional word embeddings	Labelled training data

An early strategy for the information extraction was the combination of rules (e.g. regular expressions) and features created by NLP tools. Most of those were structural features like POS tags, phrase chunks, and dependency trees. Many of these rule-based approaches (L19, L21, L23, L25, L27, L29, L30) also used gazetteer lists (i.e. lists of fixed terms to extract a specific information type). These lists are well suited for the extraction of information types with little variation. Negations, quantity units, and comparative relations are commonly used gazetteers. Some studies also created gazetteer lists to capture domain knowledge (L19, L23, L30). Nevertheless, the use of ontologies was the prevalent technique to represent domain concepts and their relations. Ontologies have the advantages of higher reusability and information density. Tools like GATE's OntoRoot Gazetteer can create term lists automatically from an ontology. Another way of combining semantic and syntactic features without the need for ontologies is the use of semantic frames or templates. The identification of such templates was performed manually in L29 and with unsupervised machine learning in L15. L29 tested one frame for the information extraction and achieved a precision of 92.3%, but it is unclear how well the frame-based information extraction performs with multiple frames.

Starting in 2019, researchers also applied deep learning to the information extraction task to address the scalability limitation arising from the rule- and ontology-based approaches. Initially, deep learning approaches required a large amount of training data. L28 and L33 show that a lack of training data can cause the results to be far from sufficient (i.e. 25.6% and 73.7% F-measure). An explanation for the performance differences is the choice of the model architecture. While L33 used a bidirectional LSTM architecture (Hochreiter & Schmidhuber, 1997), which is able to capture relationships of words in both direction and over a long distance, L28 used a simple RNN model where the information flows from left to right and declines with distance. In addition, L28 used a "none"-entity for words that do not belong to any information type, but the model did not learn to predict this entity.

L32 showed that using a Bi-LSTM model in combination with transfer learning strategies could address the lack of training data and that deep learning has the potential to boost the performance of the information extraction. With an F-measure of 87% they can outperform L28 and L33, but they are still far from the best rule- and ontology-based approaches (i.e. 95.6% and 97.9% F-measure in L25 and L27, respectively). It should be noted that L32 has an easier extraction task than L25 and L27 since the requirement units do not contain further restrictions and have a simpler sentence structure.

Transfer learning is the process of using out-of-domain training data or pretrained deep learning models and refining them for the actual task. Common transfer learning or data adaption strategies were used for various tasks:

- Data pruning to train a deep learning dependency parser (L31): In data pruning, out-of-domain data is adapted to a domain task by deleting data rows with low similarity to the development data of the target domain.

- Alternating or stage-wise training on domain and general data to improve the information extraction (L32): In the first option, the model was trained with both data sets in parallel. Only the upper layer of the model differed depending on the respective training data selection. Second, the model was trained on the general task first, then they replaced the upper layer and refined the model for the actual task.

- Pretrained word embeddings for information alignment (L38): Word embeddings and language models are a special case of stage-wise training where the model

is pretrained and has an understanding of grammar and semantics. Typical tasks for the training of language models are to predict the next word in a sentence or multiple words that were masked out. These tasks have the advantage that they can be trained in an unsupervised manner on large text corpora. Accordingly, refining a language model to a specific task requires significantly less training data than training a model from scratch (Nguyen et al., 2020).

3.7 Information transformation

In the next process step, most of the researchers postprocessed the extracted information and transformed it into intermediate formats (e.g. information tuples (L23, L25, L27), regulation trees (L20), SWRL (L22, L26), mvdXML (L22), RAINS (L21), logic statements (L30)) and further into executable representations (e.g. SPARQL (L21), XSLT (L26), Prolog logic rules (L35, L40), SQL triggers (L30)). The intermediate formats are closer to the original regulations and usually easier to read by humans. Several review papers are available that compare those representation formats based on their suitability for automated compliance checking (Solihin et al., 2019; Nawari & Alsaffar, 2015). L22 and L26 performed the conversion manually. The preceding information extraction step helped the experts to be more efficient in this process. Other approaches have already extracted all the information types that are necessary for the selected digital representation. Accordingly, they were able to transform these information entities automatically with a set of rules (L20, L21, L23, L30, L34, L35). The complexity of the rule set grows with the number of information types. L34 and L35 experimented with strategies to deal with this complexity. L34 suggests a bottom-up approach, where the clauses annotated with the semantic information elements and syntactic features are traversed and matched against a set of patterns. L35 is based on L34 and performs further experiments using the bottom-up approach. First, they transformed only the eight essential semantic information elements used in L25 (i.e. no restrictions or exceptions). They used 53 semantic mapping rules and 11 conflict resolution rules to create 1,114 logic clause elements and achieved an F-Measure of 93.8%. Second, to transform 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, and the conflict resolution rules dropped to 9. The higher information density allowed for performance improvements to 98.6% F-Measure in creating 1,936 logic clause elements.

3.8 Information alignment

With the progression towards executable formats, there is a need to align the information originating from the regulations with the information originating from the BIM or GIS in utility compliance checking. There were three different approaches: 1) Mapping regulation concepts to the building concepts, 2) mapping building concepts to the regulations, and 3) aligning the concepts and then transforming them into an executable format.

1. L26 mapped an ontology to the IFC-schema to support experts to convert the regulations from SWRL to XSLT. The rules in XSLT can then be executed to check a BIM for compliance. While L24 skipped the general alignment using an already aligned building model as a temporary solution until suitable IFC ontologies are available, they introduced an expert knowledge base with analysis steps to perform computations to retrieve advanced knowledge from the model.

2. L39 started with the conversion of design information from BIM into logic facts. These facts were then aligned with the regulations (i.e. the logic rules created in L35) using semantic transformation rules.

3. L37 aligned the extracted information elements from regulations and BIM before they were converted into logic rules and facts in L40.

Three papers focused on the techniques for semantic alignment. L36 used term-based matching and utilised WordNet (University, 2010) to be able to match synonyms. L37 looked up concepts and properties in ontologies and the buildingSMART Data Dictionary (bSDD) (BuildingSMART, 2014) and identified the final match by comparing the similarity scores (98.0% recall and 89.2% precision). In contrast, L38 used transfer learning by concatenating general word embeddings with domain word embeddings. Those embeddings were then used to encode and compare the concepts that should be aligned. Additionally, they used supervised machine learning to align the relations (e.g. spatial composition, material constituent, property) and achieved an accuracy of 77.5%.

3.9 NLP-based compliance checking

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. L39 integrated information extraction, transformation, and alignment into a unified ACCC system. L40 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 contribute an end-to-end evaluation of the frameworks. L39 achieved an F-measure of 92.8% in finding 79 non-compliant instances in a building. The information extraction and transformation from regulations were the main error sources in this example. L40 achieved 88% F-Measure to extract 24 non-compliant instances. Here, the information alignment was the primary 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 L40.

3.10 Quality assurance

The quality of the digital representations is of high importance. Automated compliance checking frameworks need a solid foundation to get acceptance. Accordingly, the translations need to be accurate and represented in a format that captures all the information included in a provision. While most studies have developed gold standards to evaluate their approaches, these data sets varied widely in size and quality. For example, L32 evaluated their semantic annotations with 30 sentences, L23 used 30 simple and 20 complex clauses to test both information extraction and transformation, and L27 used Chapter 4 of the 2012 International Energy Conservation to test the extraction of 659 information instances. In many cases, there was no information about the labelling process for the test set (L19, L23, L29, L32), some studies had one annotator and multiple reviewers (L35, L25, L10), and L27 had three annotators aiming for full annotator agreement. L36 added a manual review step to assure the quality of the extracted regulation concepts.

L41 was the only study to focus on quality assurance. They leveraged natural language generation (NLG) to recreate building code sentences from the extracted semantic information elements. The NLG metrics BLEU (Papineni et al., 2002) and ROGUE (Lin, 2004)

were used to evaluate the quality of the information elements. Both metrics measure the overlap of n-grams (unigrams and bigrams in this study). Since BLEU is precision-based and ROUGE is recall-based, these metrics complement each other well. By achieving high BLEU- and ROUGE-scores, the authors want to show that the semantic information elements can preserve all the relevant information. The achieved scores between 73% and 86% were interpreted as good comprehensibility.

Chapter 4

Gaps in research

The data extracted and categorised based on the clusters in Table 2.2 was analysed to identify eight research gaps. The primarily reflected gap categories are listed below.

- Underrepresented tasks (i.e. Table 3.1)
- Extracted information types, representation formats, and the utilised domain knowledge (i.e. Table 3.2)
- Evaluation results (i.e. Table 4.1), limitations and error sources (i.e. Chapter 3)

On the NLP task level, all the endeavours to transform regulations were performed on a set of clauses encapsulated from the original legal document. Consequently, there were no efforts to distinguish the different constituents of legal documents like definitions and general requirements. Due to the lack of benchmark data sets and agreed representation formats, an objective performance comparison of the NLP tasks is difficult. Information extraction and information alignment were the most complex and challenging tasks, where methods that are both scalable as well as high performing are still missing. Most studies were also limited to quantitative requirements. For a complete digital version of building regulations, entire regulatory documents with variously expressed requirements (e.g. text, tables, and figures) need to be translated, and high quality needs to be assured. In the following subsections, these gaps are described in detail.

Gap 1: Insufficient regulation context

Current approaches do not take advantage of clauses' full context, instead focusing on individual clauses as standalone entities. As a consequence, 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. As a step in that direction, L27 preserved the titles of the provisions, but they did not use the title information to enhance the information extraction. That is a necessity for provisions where the subject is implicit in the clause and refers to its title. Listing 4.1 shows an example from the New Zealand Standard for concrete construction, where the subject is only explicitly named in the parent provision. A future direction could combine document representation like the XML repository in L1 combined with semantic formats like SWRL or LegalRuleML. For example, Dimyadi et al. (2020) first converted regulatory documents into LegalDocML, before transforming the regulation clauses into

LegalRuleML. These XML-based formats are strongly coupled and help to maintain the connections between the semantics of regulation clauses and legal documents' structure. This connection leads to higher acceptance of the digital versions, allows easier incorporation with the rule authoring process, and makes the regulatory information more accessible for further processing. Subsequently, the contextual knowledge contained in the document can be utilised for the semantic interpretation of the regulation clauses.

```
8.1 Concrete strength at transfer
8.1.1
The drawings and specifications shall clearly define both the
specified compressive strength, f'c , and the strength at
transfer and the level of prestress required for the particular
components of the work.
```

LISTING 4.1: Example provision with subject contained in parent title (Standards New Zealand, 2011)

Gap 2: No public data sets

Currently, there are no public benchmark data sets for information extraction from building codes. The only available benchmark data sets are for text classification (L9, L12) and POS tagging (L16). Table 4.1 offers an overview of the validation results of papers about information extraction, information alignment, and end-to-end frameworks, excluding the experimental studies L13 and L34 as L25 and L35 provide more recent developments. The drawback of these results is that they only give an impression of the performance. Since they all vary in test data, extracted information types, and representation formats, a direct comparison is not possible. In addition, 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.

- Trustworthy: The data set needs to be created following common principles for open data (e.g. FAIR principles (Wilkinson et al., 2016)). To assure quality, multiple researchers and domain experts should annotate the data set in parallel. Conflicts need to be resolved by finding an agreement among all the annotators (L27).
- Diverse: The data set should consist of different regulation types (e.g. codes, standards) from several countries (e.g. IBC, European codes, NZ building codes), reflecting the differences between performance-based and prescriptive building codes, and the complexity of clauses should be balanced (L7).
- Accepted: Different researchers extracted different types of information from the regulations. An open data set would dictate what information elements need to be extracted. Accordingly, the extracted information must be suitable to represent all aspects of the original provisions.
- Open: The data set should be easily accessible to researchers. If copyright restrictions with code issuing authorities can be resolved, there are opportunities to evoke interest in the general NLP community. The depth of information extraction, the scarcity of training data, and the requirement of domain knowledge make this task a challenge for state-of-the-art NLP techniques.

ID	Evaluated task	Validation results	Test set
L18	Rule-based extraction of concept relation triplets	Precision: 70% Recall: 67% F-measure: 68% Kappa: 32	71 concept sets
L19	Rule- and ontology-based extraction of 5 information types	Precision: 92.9% Recall: 86.7% F-measure: 89.7%	30 instances
L20	Database lookup-based information extraction and transformation	53 of 83 constraints were transformable without human interference	83 constraints
L23	Rule-based information extraction and transformation	Precision: 87.9% Recall: 79.1% F-measure: 83.3%	30 simple clauses: 44 rules 20 complex clauses: 66 rules
L25	Rule- and ontology-based extraction of 10 information types	Precision: 96.9% Recall: 94.4% F-measure: 95.6%	304 sentences: 522 instances
L27	Rule- and ontology-based extraction of 9 information types	Precision: 98.5% Recall: 97.4% F-measure: 97.9%	659 instances
L28	Deep learning-based extraction of 6 information types (including NONE)	Precision: 21.1% Recall: 35.7% F-measure: 25.6%	82 sentences: 1027 instances (including 872 NONE instances)
L29	Frame-based extraction of 8 information types	Precision: 92.32%	Unknown
L30	Rule-based extraction of 6 information types and the spatial hierarchy	Precision: 95.3% Recall: 74.2% F-measure: 83.3%	50 clauses
L31	Deep learning- and rule-based extraction of requirement units 1. Requirement segmentation 2. Restriction interpretation	1. Average normalised edit distance: 0.32 2. Precision: 89% Recall: 76% F-measure: 82%	150 sentences
L32	Deep learning-based extraction of 9 information types	Precision: 88% Recall: 86% F-measure: 87%	30 sentences
L33	Deep learning-based extraction of 4 information types and classification of the relationship between procedures	Precision: 73.9% Recall: 73.9% F-measure: 73.7%	1,080 negative samples (i.e. corresponds to information type "others"); 120 annotated positive samples
L35	Rule-based information transformation	Precision: 98.2% Recall: 99.1% F-measure: 98.6%	62 sentences: 1,901 logic clause elements (i.e. 568 concepts, and 1,333 relations)
L36	Rule-based information alignment 1. Extract regulation concepts 2. IFC concept selection 3. Relationship classification	1. F-measure: 91.7% 2. Adoption Rate: 84.5% 3. Precision: 87.9%	1. 821 regulation concepts 2. 588 IFC concepts 3. 431 relations to classify
L37	Rule- and ontology-based information alignment	Precision: 98.0% Recall: 89.2%	101 instances (i.e. 47 objects and 54 properties)
L38	ML-based information alignment	Accuracy: 77.5%	80 sentences (i.e. 97 concepts and 73 relations)
L39	End-to-end	Precision: 87.6% Recall: 98.7% F-measure: 92.8%	79 non-compliant instances Requirements from chapter 19 IBC 2009
L40	End-to-end	Precision: 84.6% Recall: 91.7% F-measure: 88.0%	20 requirements, 79 design information sets, 24 non-compliant instances

Table 4.1: Validation results for information extraction, transformation, alignment and end-to-end

The data set requirements intentionally emphasise the information extraction task since the most differences were apparent in these studies. Nevertheless, the other NLP tasks like information alignment, information transformation, and document processing would benefit from a publicly available ground truth as well.

Gap 3: Agreement on complete representation requirements

The examined studies do not agree about the information required to fully represent a regulation and enable automated compliance checking. They all used a different depth of information, and most representations used for recent automated approaches are 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 without duplication, dealing with uncertainty, and the incorporation of checking methods as the main technical issues in representations.

Gap 4: Enabling scalable information extraction with exceptional performance

More research is required to enable deep information extraction, which is both scalable and high-performing. A large proportion of the information extraction approaches used a combination of rules and domain knowledge. The ontologies and gazetteer lists used as a knowledge base were developed manually or semi-automatically and often covered only sub-domains. To scale the ontology-based approaches, more effort needs to flow into the ontology development. The extraction performance (i.e. precision and recall) is strongly coupled with the quality of the ontologies. Unknown terms (L23) and implicit knowledge (L27, L30) were identified as common error sources. Also, the rules used in many of these approaches caused errors. Either some rules were missing (L23, L25, L27), complex sentence structures could not be interpreted (L23, L25), the features used in the rules were flawed (L23, L25, L27), or the rules used to resolve conflicting information elements introduced new errors (L25). One way of dealing with these limitations was to implement deep learning-based information extraction, but the performance of these approaches has not yet reached the accuracy of rule-based methods, especially when the performance of L31 and L32 are viewed end-to-end. There are two directions to progress with scalable information extraction. First, the focus could be switched to automatic ontology development to decrease the effort in formalising the domain knowledge, and rules could be developed on broader development sets. But with the increasing size and complexity of the ontologies and rule sets, problems stemming from ambiguities inherent in natural language arise, and conflicting ontology concepts and rules can lead to errors. In addition, there was only little research on automatic ontology development in the construction industry so far (Z. Zhou et al., 2016). Second, the performance of the deep learning-based extraction could be increased further by implementing state-of-the-art deep learning techniques:

- Transformer-based architecture: Much of the recent successes in NLP can be affiliated to large transformer-based models like BERT (Prasanna et al., 2020). It is likely to achieve better results for the information extraction task by leveraging this development.
- Transfer learning with language models: Pretrained versions of BERT are freely available and performed well in similar use cases. For example, (Nguyen et al., 2020)

used transfer learning with BERT to extract information from bidding documents and were able to outperform recent baselines.

- Domain knowledge: In early approaches, the use of domain knowledge and terminology was a promising method to improve the performance of NLP. The deep learning approaches for information extraction only used domain knowledge in the shape of training data. Refining language models on legal and construction documents could achieve further performance improvements. For example, L38 made use of this technique for information alignment.
- Training data generation: A big issues in many domain tasks is the sparsity of training data. Natural language generation and data augmentation could increase the amount of available training data.

It is to determine what impact such improvements can make and whether similar performance to rule- and ontology-based approaches can be achieved on a broader range of regulatory documents by intelligently combining some of the methods above.

Gap 5: Enabling scalable information alignment with exceptional performance

Similar to Gap 4, scalability issues with ontology-based methods and low performance in the machine learning-based approaches cause demand for further research on information alignment. L40 identified the information alignment as the primary error source in their end-to-end tests. Especially, super-concepts and restrictions were challenging to identify. Subsequently, they introduced an additional identification of super-concepts in L37. Furthermore, the approaches used limited sets of IFC concepts and relations. I suggest similar directions to the information extraction approach. State-of-the-art language models, refined on domain-knowledge, could be combined with the buildingSMART Data Dictionary (bSDD) and other domain knowledge sources to enhance the alignment.

Gap 6: Expanding beyond quantitative textual requirements

There is a general trend to transform only quantitative or spatial requirements into the computer-readable representation since such requirements can be used directly for compliance checks. In contrast, existential and qualitative requirements are more ambiguous and can require the definition of new checking functions like calculations or simulations. Since the structure of qualitative requirements is different, they need to be considered for the selection of a representation format and the training of an information extraction approach. In addition, a mapping mechanism from process descriptions to checking functions or the capability to perform complex reasoning tasks automatically becomes necessary.

Furthermore, 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, the corresponding provision, the table caption, the headers, the entry formats, and much more need to be taken into account. Finally, the interpretation of figures might represent the most challenging problem since they contain not only textual information but also 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

Many of the 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 (L27) and breaking down the regulations into requirement units to analyse restrictions fully and identify their relationships (L31) represent the scope of this task. Nevertheless, these efforts need to be combined in a single framework and have an F-measure close to 100% to avoid downstream errors. Especially, the qualitative characteristic of most restrictions constitutes a common source of errors in the concept alignment (L40).

Gap 8: Standardising quality assurance

Besides using test sets and L41, there was no research on quality assuring the transformed regulations. The performance requirements for the digitisation of regulations are exceptionally high. Many researchers emphasised a high recall since wrong translations could lead to non-compliant buildings and put tenants at risk. It is unknown whether an NLP approach can ever achieve a quality that will be acceptable for officials, and it is an open question of how the quality of an automated translation can be assured. As a large percentage of the effort lies in the nonrecurring, initial creation of a digital representation, I suggest keeping the human in the loop by combining the code transformation with an integrated review process. Therefore, a user interface should be added that allows the user to verify the output of the NLP models and allows to make changes where necessary. Especially information extraction and alignment should be the object of review. An advantage of such a review function is that it could allow refining a deep learning model after the actual training process using active learning. As a second step, the author would anticipate integrating the regulation transformation with the rule authoring process. Giving rule authorities a convenient method to incorporate the generation of digital codes in their process keeps the digital version up-to-date. This integration could also help to standardise regulation texts. When editors get immediate feedback that the digital representation does not correspond to the legal text or the correct IFC-concepts cannot be identified, they could clarify their formulation.

Chapter 5

Limitations

The systematic literature review was conducted rigorously to guarantee an extensive overview of the topic and allow replicable results. To the best of my knowledge, an exhaustive overview of all relevant studies was presented. Nevertheless, I would like to debate some of the decisions made for the preparation of the systematic review, recent developments in the field after the retrieval of the literature, and future directions.

First, the selection of the databases, search terms and authors can be improved. The included papers from Scopus and ProQuest were a subset of the results from Engineering Village. Accordingly, further research about the relevance of databases and journals covered by a specific database could reduce the effort to merge and exclude literature. Although some journals like "Artificial Intelligence and Law" were covered by the searched databases, consideration of legal databases could contribute to the quality of the search results. The search terms were evaluated and optimised based on two databases. Since SpringerLink searches for papers and book chapters in the full texts, it enforced the search terms to be very specific. A separate set of keywords for the full-text and abstract-and-title search could solve this problem and allow a broader search. Furthermore, the threshold for the author search was set to three to keep the effort manageable. In retrospect, a threshold of two could allow identifying further studies.

Second, a literature review captures the state-of-the-art at a certain point in time. Between the literature retrieval and the final documentation of the review passed some time, and the database alerts, which were set up with the original search queries, pointed out additional literature. Three of those papers should be mentioned here to give the reader a complete picture.

1. Moon et al. (2021) provides an update to their preliminary results in L28. They replaced the previously used RNN with a Bi-LSTM for named entity recognition and improved their results significantly (i.e. 91.9% precision and 91.4% recall). They collected 4,659 sentences from 56 construction specifications labelled with organisation, action, element, standard, and reference. This data set is available on request.

2. J. Song et al. (2020) suggests the extraction of a predicate-argument structure from building requirements. Therefore, they adapted semantic role labels to represent building regulations. They achieved an average of 49% precision and 65% recall for the extraction of ten semantic roles with a Bi-LSTM model and 350 annotated sentences, split into training, validation and test sets. Finally, they revealed their

plan to integrate the rule conversion with the rule-authoring process and implement a user interface to review the rule translation (i.e. Gap 8).

3. F. Li et al. (2020) manage the complexity of regulations by extracting triplets consisting of subject, predicate and object. Three domain experts annotated 1,320 clauses from 14 Chinese building codes. They developed a deep learning-based framework that made use of Bi-LSTMs, self-attention, and character-level as well as word-level embeddings. They outperformed various baseline architectures with 88.1% precision and 85.2% recall.

These papers confirm the potential of information extraction with deep learning (i.e. Gap 5). Also, further steps towards public data sets (i.e. Gap 3) were made by Moon et al. (2021). Nevertheless, due to the different context of the study, the extracted information types differ from information types that were commonly extracted for automated compliance checking. J. Song et al. (2020) and F. Li et al. (2020) introduce new methods to represent regulations attributing to Gap 4, an agreed representation that can capture the complexity of building regulations is yet to be found. Moreover, both representations are not limited to quantitative requirements (i.e. Gap 7) and improve the representation of complex regulations (i.e. Gap 8). The approach of F. Li et al. (2020) is comparable to L18, but the predicates are adapted to represent relations common to compliance checking, and the subjects' functions were determined. By successfully leveraging self-attention, they move towards transformer-based information extraction (i.e. Gap 5). Additionally, J. Song et al. (2020) refined general word embeddings to included domain-specific vocabulary.

Third, the literature review was conducted with a specialised domain scope, excluding the use of NLP for out-of-domain legal texts. Generally, NLP has a greater research interest in the legal domain, and many of the aforementioned problems appear there in a similar form. Hence, it would be beneficial to complement the presented results with a broader literature review. Legal document processing and information extraction from legal documents could be of particular interest.

Chapter 6

Conclusion

This systematic literature review identified 41 relevant articles about the interpretation of building regulations using NLP. These articles were selected from 1,962 records retrieved from six databases, plus further candidate articles detected by the backwards snowballing and author search strategies. The articles were then categorised, summarised, and analysed. Finally, eight research gaps were identified, and recommendations how to address those gaps were provided.

The regulation computerisation process was commonly performed by extracting information elements from the regulations, transforming these elements into a computer-readable format and aligning the information elements with IFC-concepts and relations. In addition, some authors proposed text classification and syntactic and semantic text analysis to support the process. Initially, the information extraction was performed with pattern-based rules, structural text features, and formalised domain knowledge like gazetteer lists or ontologies. These approaches performed very well in many studies, but they are widely considered to have low scalability and high reliance on the quality of the rules and knowledge base. Machine learning has been explored to fill the gap, but these studies have not reached the high performance of rule-based approaches yet. The scarcity of training data, the lack of open data sets, and the disagreement about the requirements of a representation for building regulations are hindrances to rapid improvement. Also, most approaches were limited to quantitative requirements. Qualitative and existential requirements, as well as requirements in the form of tables and figures, need to be converted into a computable format as well to reduce the gap to fully automated compliance checking. Using state-of-the-art NLP and integrating the transformation process with appropriate, potentially manual quality assurance measures could help to close some of these gaps.

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

Agnoloni, T., & Tiscornia, D. (2010). Semantic Web Standards and Ontologies for Legislative Drafting Support. In *Electronic participation.* doi: 10.1007/978-3-642-15158-3_16

Al Qady, M., & Kandil, A. (2010). Concept relation extraction from construction documents using natural language processing. *Journal of Construction Engineering and Management*, *136*(3), 294–302. doi: 10.1061/(ASCE)CO.1943-7862.0000131

Al Qady, M., & Kandil, A. (2015). Automatic classification of project documents on the basis of text content. *Journal of Computing in Civil Engineering*, *29*(3). doi: 10.1061/(ASCE)CP.1943-5487.0000338

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

BIM Acceleration Committee. (2019). *The New Zealand BIM Handbook* (v3.1 ed.).

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit.* " 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*.

Brüninghaus, S., & Ashley, K. D. (2001). Improving the Representation of Legal Case Texts with Information Extraction Methods 1 Motivation: Indexing and Information Extraction for Legal Cases. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.9048&rep=rep1&type=pdf

BuildingSMART. (2014). *buildingsmart data dictionary.* http://bsdd.buildingsmart.org/. (Accessed: 2021-05-03)

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

Cerovsek, T., Gudnason, G., & Lima, C. (2006). An European network of decentralized portals enabling e-business with building regulations - The CONNIE project. In *Proceedings of the 6th european conference on product and process modelling - ework and ebusiness in architecture, engineering and construction, ecppm 2006* (pp. 561–570). Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-61849149404&partnerID=40&md5=8b5191635b6c70bd79b8832cc1fb1de6

Chalkidis, I., & Kampas, D. (2019, 6). Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, *27*(2), 171–198. doi: 10.1007/s10506-018-9238-9

Cheng, C. P., Lau, G. T., Law, K. H., Pan, J., & Jones, A. (2008, 9). Regulation retrieval using industry specific taxonomies. *Artificial Intelligence and Law*, *16*(3), 277–303. doi: 10.1007/s10506-008-9065-5

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2015). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *arXiv preprint arXiv:1409.1259*, 103–111. doi: 10.3115/v1/w14-4012

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*, 4171–4186.

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*, 39–46.

El-Diraby, T. A., Lima, C., & Feis, B. (2005, 10). Domain Taxonomy for Construction Concepts: Toward a Formal Ontology for Construction Knowledge. *Journal of Computing in Civil Engineering*, *19*(4), 394–406. Retrieved from http://ascelibrary.org/doi/10.1061/%28ASCE%290887-3801%282005%2919%3A4%28394%29 doi: 10.1061/(ASCE)0887-3801(2005)19:4(394)

El-Diraby, T. E. (2013, 7). Domain ontology for construction knowledge. *Journal of Construction Engineering and Management*, *139*(7), 768–784. doi: 10.1061/(ASCE)CO.1943-7862.0000646

El-Gohary, N. M., & El-Diraby, T. E. (2010, 7). Domain ontology for processes in infrastructure and construction. *Journal of Construction Engineering and Management*, *136*(7), 730–744. doi: 10.1061/(ASCE)CO.1943-7862.0000178

Emani, C., Silva, C. F. d., Fiès, B., Zarli, A., & Ghodous, P. (2016). *An Approach for Automatic Formalization of Business Rules.* hal.archives-ouvertes.fr. Retrieved from https://hal.archives-ouvertes.fr/hal-01396916/

Fahad, M., Bus, N., & Andrieux, F. (2016). Towards mapping certification rules over BIM. In *Proceedings of the 33rd international conference of cib w78, brisbane, australia, 31 october - 2 november (issn: 2706-6568).* Retrieved from https://itc.scix.net/pdfs/w78-2016-paper-001.pdf

Fenves, S. J. (1966). Tabular Decision Logic for Structural Design. *Journal of the Structural Division*, *92*(6), 473–490.

Fiatech Regulatory Streamlining Committee. (2012). AutoCodes Project: Phase 1, Proof-of-Concept Final Report. (March), 20.

Giuda, G. M. D., Locatelli, M., Schievano, M., Pellegrini, L., Pattini, G., Giana, P. E., & Seghezzi, E. (2020). *Natural Language Processing for Information and Project Management.* Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-33570-0_9

Hassan, F. U., & Le, T. (2020, 5). Automated Requirements Identification from Construction Contract Documents Using Natural Language Processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *12*(2). doi: 10.1061/(ASCE)LA.1943-4170.0000379

Hjelseth, E. (2012). Converting performance based regulations into computable rules in BIM based model checking software. In *ework and ebusiness in architecture, engineering and construction - proceedings of the european conference on product and process modelling 2012, ecppm 2012* (pp. 461–469). Department of Mathematical Sciences and Technology, Norwegian University of Life Sciences (UMB), Norway. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84863528634&partnerID=40&md5=db67f29b3240ae633146b10dca1a57bd

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. Retrieved from https://doi.org/10.1162/neco.1997.9.8.1735 doi: 10.1162/neco.1997.9.8.1735

Kim, T., & Chi, S. (2019, 3). Accident Case Retrieval and Analyses: Using Natural Language Processing in the Construction Industry. *Journal of Construction Engineering and Management*, *145*(3). doi: 10.1061/(ASCE)CO.1943-7862.0001625

Kitchenham, B. (2004). Procedures for Performing Systematic Literature Reviews. *Joint Technical Report, Keele University TR/SE-0401 and NICTA TR-0400011T.1*, 33.

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

Lau, G. T., & Law, K. H. (2004). An Information Infrastructure for Comparing Accessibility Regulations and Related Information from Multiple Sources. In *Proceedings of the 10th international conference on computing in civil and building engineering, weimar, germany, june 2-4* (pp. 1–11). Retrieved from http://eil.stanford.edu/publications/gloria_lau/icccbe.pdf

Lau, G. T., Law, K. H., & Wiederhold, G. (2006, 12). A relatedness analysis of government regulations using domain knowledge and structural organization. *Information Retrieval*, *9*(6), 657–680. doi: 10.1007/s10791-006-9010-8

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. In *Advances in informatics and computing in civil and construction engineering.* doi: 10.1007/978-3-030-00220-6_68

Lee, J., Yi, J. S., & Son, J. (2019, 5). Development of Automatic-Extraction Model of Poisonous Clauses in International Construction Contracts Using Rule-Based NLP. *Journal of Computing in Civil Engineering*, *33*(3). doi: 10.1061/(ASCE)CP.1943-5487.0000807

Li, F., Song, Y., & Shan, Y. (2020). Joint extraction of multiple relations and entities from building code clauses. *Applied Sciences (Switzerland)*, *10*(20), 1–18. doi: 10.3390/app10207103

Li, S., Cai, H., & Kamat, V. R. (2016, 12). Integrating Natural Language Processing and Spatial Reasoning for Utility Compliance Checking. *Journal of Construction Engineering and Management*, *142*(12), 04016074. Retrieved from http://ascelibrary.org/doi/10.1061/%28ASCE%29CO.1943-7862.0001199 doi: 10.1061/(ASCE)CO.1943-7862.0001199

Li, Z., Zhao, H., Wang, R., & Parnow, K. (2020). *High-order Semantic Role Labeling.* Retrieved from http://arxiv.org/abs/2010.04641

Liang, V.-C., & Garrett, J. H. (2000, 4). Java-Based Regulation Broker. *Journal of Computing in Civil Engineering*, *14*(2), 100–108. Retrieved from http://ascelibrary.org/doi/10.1061/%28ASCE%290887-3801%282000%2914%3A2%28100%29 doi: 10.1061/(ASCE)0887-3801(2000)14:2(100)

Lima, C., Silva, C. F. d., & Pimentão, J. P. (2006). Assessing the Quality of Mappings Between Semantic Resources in Construction. In *Intelligent computing in engineering and architecture.* Retrieved from http://link.springer.com/chapter/10.1007/11888598_38

Lin, C.-Y. (2004, 7). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text summarization branches out* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W04-1013

Liu, K., & El-Gohary, N. (2016). Ontology-based Sequence Labelling for Automated Information Extraction for Supporting Bridge Data Analytics. In *Procedia engineering* (Vol. 145, pp. 504–510). Elsevier Ltd. doi: 10.1016/j.proeng.2016.04.035

Lv, X., & El-Gohary, N. M. (2016, 11). Semantic Annotation for Supporting Context-Aware Information Retrieval in the Transportation Project Environmental Review Domain. *Journal of Computing in Civil Engineering*, *30*(6). doi: 10.1061/(ASCE)CP.1943-5487.0000565

Mahdavi, A., & Taheri, M. (2018). A building performance indicator ontology. In *ework and ebusiness in architecture, engineering and construction - proceedings of the 12th european conference on product and process modelling, ecppm 2018* (p. 385). Department of Building Physics and Building Ecology, TU Wien, Vienna, Austria. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85071845285&doi=10.1201%2F9780429506215-48&partnerID=40&md5=b8a53c5da977fa60430d499fc0b01965 doi: 10.1201/9780429506215-48

Mahfouz, T., Kandil, A., & Davlyatov, S. (2018). Identification of latent legal knowledge in differing site condition (DSC) litigations. *Automation in Construction*, *94*, 104–111. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049071070&doi=10.1016%2Fj.autcon.2018.06.011&partnerID=40&md5=a06e01297329f966b24c890f061a0a97 doi: 10.1016/j.autcon.2018.06.011

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

Mastrodonato, C., De Ferrari, A., Cioffi, M., Bourdeau, M., & Zarlì, A. (2010). An information platform for the European construction sector. In *echallenges e-2010 conference.* D'Appolonia SpA, Via San Nazaro, Geneva, 16145, Italy. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-79957492216&partnerID=40&md5=d9cb3858c4e66b77e63d266c99c75b80

Mathot, M., Coenders, J., & Rolvink, A. (2016). Feasibility of a Knowledge-Based Engineering framework for the AEC industries. *Proceedings of IASS Annual Symposia*, *2016*(13), 1–9. Retrieved from https://www.ingentaconnect.com/content/iass/piass/2016/00002016/00000013/art00004

McGibbney, L. J., & Kumar, B. (2013). An intelligent authoring model for subsidiary legislation and regulatory instrument drafting within construction and engineering industry. *Automation in construction*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580513000502

McGibbney, L. J., & Kumar, B. (2015, 1). A framework for regulatory ontology construction within AEC domain. In *Ontology in the aec industry: A decade of research and development in architecture, engineering, and construction* (pp. 193–216). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784413906.ch09

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 1–9.

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

MIREL. (2017). Collection of state-of-the-art NLP tools for processing of legal text. (690974), 1–22. Retrieved from http://www.mirelproject.eu/

Moon, S., Lee, G., Chi, S., & Oh, H. (2019, 6). Automatic Review of Construction Specifications Using Natural Language Processing. In (pp. 401–407). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784482438.051

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), 04020147. Retrieved from http://ascelibrary.org/doi/10.1061/%28ASCE%29CO.1943-7862.0001953 doi: 10.1061/(ASCE)CO.1943-7862.0001953

Moon, S., Shin, Y., Hwang, B.-G., & Chi, S. (2018, 12). Document Management System Using Text Mining for Information Acquisition of International Construction. *KSCE Journal of Civil Engineering*, *22*(12), 4791–4798. doi: http://dx.doi.org/10.1007/s12205-018-1528-y

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

Nawari, N. O., & Alsaffar, A. (2015). *Understanding computable building codes.* pdfs.semanticscholar.org. Retrieved from https://pdfs.semanticscholar.org/2a7d/8093e9eb1568b2b26409b4fa8b8cece7ab07.pdf

Nguyen, M. T., Phan, V. A., Linh, L. T., Son, N. H., Dung, L. T., Hirano, M., & Hotta, H. (2020). Transfer Learning for Information Extraction with Limited Data. *Communications in Computer and Information Science*, *1215 CCIS*, 469–482. doi: 10.1007/978-981-15-6168-9_38

Niemeijer, R. A., De Vries, B., & Beetz, J. (2014). Freedom through constraints: User-oriented architectural design. *Advanced Engineering Informatics*, *28*(1), 28–36. Retrieved from http://dx.doi.org/10.1016/j.aei.2013.11.003 doi: 10.1016/j.aei.2013.11.003

Niu, J., Issa, R. R., & Mutis, I. (2015, 1). Taxonomy development toward the domain ontology of construction contracts: A case study on AIA A201-2007. In *Ontology in the aec industry: A decade of research and development in architecture, engineering, and construction* (pp. 217–250). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784413906.ch10

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002, 7). Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P02-1040 doi: 10.3115/1073083.1073135

Prasanna, S., Rogers, A., & Rumshisky, A. (2020). When BERT Plays the Lottery, All Tickets Are Winning. Retrieved from http://arxiv.org/abs/2005.00561

Preidel, C., & Borrmann, A. (2018). BIM-based code compliance checking. In *Building information modeling: Technology foundations and industry practice* (pp. 367–381). Chair of Computational Modeling and Simulation, Technical University of Munich, München, Germany. doi: 10.1007/978-3-319-92862-3_22

Riloff, E., & Phillips, W. (2004). An Introduction to the Sundance and AutosSlog Systems. *Technical Report UUCS-04-015, School of Computing, University of Utah*(UUCS-04- 015), 1–47.

Roshnavand, A. A., Nik-Bakht, M., & Han, S. H. (2019). Towards Automated Analysis of Ambiguity in Modular Construction Contract Documents (A Qualitative & Quantitative Study). In *Advances in informatics and computing in civil and construction engineering.* doi: 10.1007/978-3-030-00220-6_41

Salama, D. A., & El-Gohary, N. M. (2013, 11). Automated Compliance Checking of Construction Operation Plans Using a Deontology for the Construction Domain. *Journal of Computing in Civil Engineering*, *27*(6), 681–698. Retrieved from http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000298 doi: 10.1061/(ASCE)CP.1943-5487.0000298

Salama, D. M., & El-Gohary, N. M. (2016, 1). *Semantic Text Classification for Supporting Automated Compliance Checking in Construction* (Vol. 30) (No. 1). American Society of Civil Engineers (ASCE). doi: 10.1061/(ASCE)CP.1943-5487.0000301

Seo, P. H., Lin, Z., Cohen, S., Shen, X., & Han, B. (2016). Hierarchical Attention Networks. *ArXiv*, 1480–1489. Retrieved from http://arxiv.org/abs/1606.02393

Shi, L., & Roman, D. (2017). From standards and regulations to executable rules: A case study in the Building Accessibility domain. In *Ceur workshop proceedings* (Vol. 1875). Statsbygg, Pb. 8106 Dep., Oslo, 0032, Norway. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85027883114&partnerID=40&md5=1e71c6dbce0fe118ff6590001a23220c

Solihin, W., Dimyadi, J., & Lee, Y.-C. (2019). In Search of Open and Practical Language-Driven BIM-Based Automated Rule Checking Systems. In *Advances in informatics and computing in civil and construction engineering* (pp. 577–584). Springer International Publishing. doi: 10.1007/978-3-030-00220-6_69

Song, J., Kim, J., & Lee, J.-K. (2018). NLP and Deep Learning-based Analysis of Building Regulations to support Automated Rule Checking System. In *Isarc proceedings of the international symposium on automation and robotics in construction* (Vol. 35, pp. 1–7). Department of Interior Architecture & Built Environment, Yonsei University, Republic of Korea: IAARC Publications.

Song, J., Lee, J.-k., Choi, J., & Kim, I. (2020). Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences. *Journal of Computational Design and Engineering*, *7*(0), 1–14. doi: 10.1093/jcde/qwaa046

Song, J. Y., Kim, J. S., Kim, H., Choi, J., & Lee, J. K. (2018). Approach to capturing design requirements from the existing architectural documents using natural language processing technique. In *Caadria 2018 - 23rd international conference on computer-aided architectural design research in asia: Learning, prototyping and adapting* (Vol. 2, pp. 247–254). Department of Interior Architecture and Built Environment, Yonsei University, South Korea. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85056116178&partnerID=40&md5=d5394ca360b69eaaa7f4688c98ac5906

Standards New Zealand. (2011). *NZS 3109:1997 New Zealand Standard - Concrete construction* (No. Incorporating Amendments No.1 and No. 2).

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

Tang, S., & Golparvar-Fard, M. (2017, 6). Joint Reasoning of Visual and Text Data for Safety Hazard Recognition. In (pp. 450–457). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784480847.056

University, P. (2010). *Wordnet.* Princeton University "About WordNet.".

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*, 1–9. doi: 10.1038/sdata.2016.18

Xu, X., & Cai, H. (2019). Semantic Frame-Based Information Extraction from Utility Regulatory Documents to Support Compliance Checking. In *Advances in informatics and computing in civil and construction engineering.* doi: 10.1007/978-3-030-00220-6_27

Xu, X., Cai, H., & Chen, K. (2019). Modeling 3D Spatial Constraints to Support Utility Compliance Checking. In *Computing in civil engineering 2019: Visualization, information modeling, and simulation - selected papers from the asce international conference on computing in civil engineering 2019* (pp. 439–446). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784482421.056

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*, 1–10. doi: 10.1061/(ASCE)CP.1943-5487.0000917

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*, 1384p. doi: https://doi.org/10.1061/9780784482865.053

Youssef, A., Osman, H., Georgy, M., & Yehia, N. (2018, 5). Semantic Risk Assessment for Ad Hoc and Amended Standard Forms of Construction Contracts. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *10*(2). doi: 10.1061/(ASCE)LA.1943-4170.0000253

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. In *Proceedings*

- *14th international conference on computational intelligence and security, cis 2018* (pp. 277–280). Chinese Flight Test Establishment, Xi'an, 710089, China. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060735147&doi=10.1109%2FCIS2018.2018.00068&partnerID=40&md5=65fee1aa7fa2691cce0b8c8a9161d568 doi: 10.1109/CIS2018.2018.00068

Zhang, J., & El-Gohary, N. (2012). Extraction of construction regulatory requirements from textual documents using natural language processing techniques. In *Congress on computing in civil engineering, proceedings* (pp. 453–460). doi: 10.1061/9780784412343.0057

Zhang, J., & El-Gohary, N. M. (2013). *Handling sentence complexity in information extraction for automated compliance checking in construction.* architektur-informatik.scix.net. Retrieved from http://architektur-informatik.scix.net/pdfs/w78-2013-paper-89.pdf

Zhang, J., & El-Gohary, N. M. (2015). Automated information transformation for automated regulatory compliance checking in construction. In (Vol. 29). Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana; IL; 61801, United States: American Society of Civil Engineers (ASCE). Retrieved from http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000427 doi: 10.1061/(ASCE)CP.1943-5487.0000427

Zhang, J., & El-Gohary, N. M. (2016a, 9). Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques. *Journal of Computing in Civil Engineering*, *30*(5). doi: 10.1061/(asce)cp.1943-5487.0000536

Zhang, J., & El-Gohary, N. M. (2016b, 3). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering*, *30*(2). doi: 10.1061/(ASCE)CP.1943-5487.0000346

Zhang, J., & El-Gohary, N. M. (2017, 1). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction*, *73*, 45–57. Retrieved from http://dx.doi.org/10.1016/j.autcon.2016.08.027 doi: 10.1016/j.autcon.2016.08.027

Zhang, L., & El-Gohary, N. M. (2016a, 3). Epistemology-Based Context-Aware Semantic Model for Sustainable Construction Practices. *Journal of Construction Engineering and Management*, *142*(3). doi: 10.1061/(ASCE)CO.1943-7862.0001055

Zhang, L., & El-Gohary, N. M. (2016b, 3). Epistemology-Based Context-Aware Semantic Model for Sustainable Construction Practices. *Journal of Construction Engineering and Management*, *142*(3). doi: 10.1061/(ASCE)CO.1943-7862.0001055

Zhang, L., & Issa, R. R. (2011). IFC-based construction industry ontology and semantic web services framework. In *Congress on computing in civil engineering, proceedings* (pp. 657–664). doi: 10.1061/41182(416)81

Zhang, R., & El-Gohary, N. (2018). A clustering approach for analyzing the computability of building code requirements. In *Construction research congress 2018: Construction information technology - selected papers from the construction research congress 2018* (Vol. 2018-April, pp. 86–95). doi: 10.1061/9780784481264.009

Zhang, R., & El-Gohary, N. (2019a). A Machine Learning Approach for Compliance Checking-Specific Semantic Role Labeling of Building Code Sentences. In *Advances in informatics and computing in civil and construction engineering* (pp. 561–568). Springer International Publishing. doi: 10.1007/978-3-030-00220-6_67

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. In *Recent research in sustainable structures* (pp. 64–73). Springer International Publishing. doi: 10.1007/978-3-030-34216-6_5

Zhang, R., & El-Gohary, N. (2019c). A machine learning-based method for building code requirement hierarchy extraction. In *Proceedings, annual conference - canadian society for civil engineering* (Vol. 2019-June). University of Illinois at Urbana-Champaign, United States. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-85080874755&partnerID=40&md5=c23b99327322394004bb7b35f2718da5

Zhang, R., & El-Gohary, N. (2019d, 6). Unsupervised Machine Learning for Augmented Data Analytics of Building Codes. In (pp. 74–81). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784482438.010

Zhang, R., & El-Gohary, N. (2020a). A Deep-Learning Method for Evaluating Semantically-Rich Building Code Annotations. In *Eg-ice 2020 workshop on intelligent computing in engineering, proceedings* (pp. 285–293).

Zhang, R., & El-Gohary, N. (2020b). A Machine-Learning Approach for Semantically-Enriched Building-Code Sentence Generation for Automatic Semantic Analysis. *Proceedings of the 2020 ASCE Construction Research Congress (CRC), Tempe, AZ, March 08-10, 2020*, 1–10.

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*. Retrieved from http://dx.doi.org/10.1016/j.aei.2019.101003 doi: 10.1016/j.aei.2019.101003

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). doi: 10.1061/(ASCE)CP.1943-5487.0000513

Zhou, P., & El-Gohary, N. (2016b, 7). Ontology-based multilabel text classification of construction regulatory documents. *Journal of Computing in Civil Engineering*, *30*(4). doi: 10.1061/(ASCE)CP.1943-5487.0000530

Zhou, P., & El-Gohary, N. (2017, 2). Ontology-based automated information extraction from building energy conservation codes. *Automation in Construction*, *74*, 103–117. Retrieved from http://dx.doi.org/10.1016/j.autcon.2016.09.004 doi: 10.1016/j.autcon.2016.09.004

Zhou, P., & El-Gohary, N. (2018a). Automated matching of design information in BIM to regulatory information in energy codes. In *Construction research congress 2018: Construction information technology - selected papers from the construction research congress 2018* (Vol. 2018-April, pp. 75–85). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784481264.008

Zhou, P., & El-Gohary, N. (2018b). Text and Information Analytics for Fully Automated Energy Code Checking. *International Congress and Exhibition "Sustainable Civil Infrastructures: Innovative Infrastructure Geotechnology"*, 196–208. Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-01905-1_11

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, 207–212. doi: 10.18653/v1/p16-2034

Zhou, Z., Goh, Y. M., & Shen, L. (2016, 11). Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry. *Journal of Computing in Civil Engineering*, *30*(6). doi: 10.1061/(ASCE)CP.1943-5487.0000594

Appendix A

Search queries

A.1 Engineering Village

- Query: (process* "natural language" OR "natural language understanding" OR NLP OR "semantic-based" OR "text analysis" OR "text processing" OR "information extraction" OR "information retrieval" OR "text classification") AND ("building code" OR "building codes" OR "building standard" OR "building standards" OR "construction code" OR "construction codes" OR "building regulation" OR "building regulations" OR "construction regulation" OR "construction regulations" OR ((regulation OR regulatory) AND ("AEC industry" OR "construction industry" OR "building industry" OR "AEC domain" OR "construction domain" OR "building domain" OR "construction sector" OR "building sector" OR "AEC sector" OR "civil engineering")))
- Adjustments: Adding controlled vocabulary (semantics, classification (of information))
- Databases: Compendex, Inspec & Knovel
- Scope: Search anywhere
- Limits: 2000 - 27 April 2020
- Initial records: 254
- Steps: Remove duplicates with preference Compendex
- **217 database records**

A.2 ASCE

- Query: (process* "natural language" OR "natural language understanding" OR NLP OR "semantic-based" OR "text analysis" OR "text processing" OR "information extraction" OR "information retrieval" OR "text classification") AND ("building code" OR "building codes" OR "building standard" OR "building standards" OR "construction code" OR "construction codes" OR "building regulation" OR "building regulations" OR "construction regulation" OR "construction regulations"

OR ((regulation OR regulatory) AND ("AEC industry" OR "construction industry" OR "building industry" OR "AEC domain" OR "construction domain" OR "building domain" OR "construction sector" OR "building sector" OR "AEC sector" OR "civil engineering")))

- Scope: Search anywhere
- Limits: 2000 - 27 April 2020
- Initial records: 274
- Steps: Manual retrieval with Mendeley add-on; Exclude Front Matter (18) and Back Matter(2)
- **254 database records**

A.3 SpringerLink

- Query: ((process* NEAR "natural language") OR "natural language understanding" OR NLP OR "semantic-based" OR "text analysis" OR "text processing" OR "information extraction" OR "information retrieval" OR "text classification") AND ("building code" OR "building codes" OR "building standard" OR "building standards" OR "construction code" OR "construction codes" OR "building regulation" OR "building regulations" OR "construction regulation" OR "construction regulations" OR ((regulation* OR regulatory) AND ("AEC industry" OR "construction industry" OR "building industry" OR "AEC domain" OR "construction domain" OR "building domain" OR "construction sector" OR "building sector" OR "AEC sector" OR "civil engineering")))
- Adjustments: regulation*
- Scope: Search anywhere
- Retrieved at 28 April
- Initial records: 803
- Steps:
  - Retrieve records per discipline (Computer Science (239), Engineering(168))
  - Convert csv to ris with python script
  - Automatic duplicate removal via Mendeley (2)
  - 405 Records in Mendeley
  - Manually remove records published before 2000
- **314 database records**

A.4 ProQuest

- Query: noft(((process* NEAR "natural language") OR "natural language understanding" OR NLP OR semantic OR "text analysis" OR "text processing" OR "information extraction" OR "information retrieval" OR "text classification") AND ("building code" OR "building codes" OR "building standard" OR "building standards" OR "construction code" OR "construction codes" OR "building regulation" OR "building regulations" OR "construction regulation" OR "construction regulations" OR ((regulation OR regulatory) AND ("AEC industry" OR "construction industry" OR "building industry" OR "AEC domain" OR "construction domain" OR "building domain" OR "construction sector" OR "building sector" OR "AEC sector" OR "civil engineering"))))
- Adjustments: Match subject terms (i.e. semantic-based -> semantics)
- Scope: noft (i.e. no full text)
- Databases: All
- Limits: 2000 - 28 April 2020
- Initial records: 141
- Steps: Automatic duplicate removal via Mendeley (19)
- **122 database records**

A.5 Scopus

- Query: TITLE-ABS-KEY (((process* W/10 "natural language") OR "natural language understanding" OR nlp OR semantic OR "text analysis" OR "text processing" OR "information extraction" OR "information retrieval" OR "text classification" OR "classification (of information)") AND ("building code" OR "building codes" OR "building standard" OR "building standards" OR "construction code" OR "construction codes" OR "building regulation" OR "building regulations" OR "construction regulation" OR "construction regulations" OR ((regulation OR regulatory OR "laws and legislation") AND ("AEC industry" OR "construction industry" OR "building industry" OR "AEC domain" OR "construction domain" OR "building domain" OR "construction sector" OR "building sector" OR "AEC sector" OR "civil engineering")))) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI"))
- Adjustments: process* W/10 "natural language"; Keywords added: semantic, "classification (of information)", "laws and legislation"
- Limits: 2000 - 28 April 2020, Subject areas (Engineering, Computer Science)
- **130 database records**

A.6 Google Scholar

- Query: "building codes" OR "building code" OR "building standards" OR "construction regulations" OR "construction regulation" OR "building regulations" "natural language processing" -"source code" -"software engineering" -sociology -telecommunication
- Adjustments:
  - Query length restriction -> Focus on main search terms
  - Add restrictive terms to remove clusters of out-of-domain records
- Limits: 2000 - 28 April 2020
- 360 Initial records
- Steps:
  - Automatic retrieval with "Publish and Perish"-tool
  - Automatic duplicate removal via Mendeley (4)
- **356 database records**