A Machine-Learning Approach for Semantically-Enriched Building-Code Sentence Generation for Automatic Semantic Analysis

Ruichuan ZHANG[1] and Nora EL-GOHARY[2]

[1] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, IL, 61801; PH (217) 979-0620; email: rzhang65@illinois.edu

[2] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, IL, 61801; PH (217) 333-6620; email: gohary@illinois.edu

ABSTRACT

Existing automated code checking (ACC) systems require the extraction of requirements from regulatory textual documents into computer-processable rule representations. The information extraction processes in those ACC systems are based on either human interpretation, manual annotation, or predefined automated information extraction rules. Despite the high performance they showed, rule-based information extraction approaches, by nature, lack sufficient scalability – the rules typically need some level of adaptation if the characteristics of the text change. Machine learning-based methods, instead of relying on hand-crafted rules, automatically capture the underlying patterns of the existing training text and have a great capability of generalizing to a variety of texts. A more scalable, machine learning-based approach is thus needed to achieve a more robust performance across different types of codes/documents for automatically generating semantically-enriched building-code sentences for the purpose of ACC. To address this need, this paper proposes a machine learning-based approach for generating semantically-enriched building-code sentences, which are annotated syntactically and semantically, for supporting IE. For improved robustness and scalability, the proposed approach uses transfer learning strategies to train deep neural network models on both general-domain and domain-specific data. The proposed approach consists of four steps: (1) data preparation and preprocessing; (2) development of a base deep neural network model for generating semantically-enriched building-code sentences; (3) model training using transfer learning strategies; and (4) model evaluation. The proposed approach was evaluated on a corpus of sentences from the 2009 International Building Code (IBC) and the Champaign 2015 IBC Amendments. The preliminary results show that the proposed approach achieved an optimal precision of 88%, recall of 86%, and F1-measure of 87%, indicating good performance.

INTRODUCTION

Existing automated code checking (ACC) systems require the extraction of requirements from regulatory textual documents into computer-processable rule representations. The information extraction (IE) processes in those ACC systems rely on either human interpretation, manual annotation, or predefined automated information extraction rules. For example, the state-of-the-art methods for extracting ACC-related information are rule-based (e.g., Zhang and El-Gohary 2013, Zhou and

El-Gohary 2017), which require human effort to develop rules for automatically extracting the information from the building codes. Despite the high performance they showed, rule-based approaches, by nature, lack sufficient scalability – the rules typically need some level of adaptation if the characteristics of the text change. Machine learning-based methods, instead of relying on hand-crafted rules, automatically capture the underlying patterns of the existing training text and have a great capability of generalizing to a variety of texts. A more scalable, machine learning-based approach is thus needed to achieve a more robust performance – without requiring manual rule adaptation effort – across different types of codes/documents for automatically generating semantically-enriched building-code sentences for the purpose of ACC.

To address this need, this paper proposes a machine learning-based approach for generating such semantically-enriched building-code sentences that are annotated with semantic information elements (Zhang and El-Gohary 2013) and syntactic fillers, and thus are ready for computer processing and reasoning. The proposed approach uses transfer learning strategies to train deep neural network models on both general-domain and domain-specific data. On one hand, general-domain data are large-scale and pattern-rich, which helps train the model to deal with different text patterns across multiple codes/documents for increased robustness and scalability; but general-domain data are relatively different from the domain-specific data in terms of vocabularies, syntactics, and semantics. On the other hand, domain-specific data (i.e., annotated building code sentences) are the target data from the architecture/engineering/construction (AEC) domain, but they are much smaller in size and are lower in syntactic and semantic richness, which would limit the robustness and scalability of the deep neural network model if they are solely used for training. The proposed approach, thus, takes the best of both worlds.

The proposed approach consists of four main steps: (1) prepare and preprocess training and testing data from both outside of the AEC domain (i.e., the general-domain data) and within the AEC domain (i.e., the domain-specific data); (2) development of a base deep neural network model for generating semantically-enriched building-code sentences; (3) model training using different transfer learning strategies; and (4) model evaluation using precision, recall, and F1-measure.

BACKGROUND

Semantic Text Enrichment

Semantic text enrichment aims to attach computer-processible semantic information to the natural language text (Abel et al. 2011). Compared to the original natural language text, the semantically-enriched text contains highly-structured, and often domain-specific semantic information that can be used directly by computers for semantic analysis tasks. There are many applications, tools, and platforms for creating and managing semantically-enriched texts (e.g., semantic wiki). And many research efforts have been focused on automating the process of generating semantically-enriched text and/or text semantic annotation (e.g., Abel et al. 2011, Dugas et al. 2016). To solve the needs for automated compliance checking of building designs, different types of building-code requirement representations have been proposed and can be

potentially used as the semantic annotations in the semantically-enriched building-code sentences, such as the semantic information elements (Zhang and El-Gohary 2013), shown in Table 1.

Table 1. Essential Semantic Information Elements for Representing Requirements for Compliance Checking Purposes (Zhang and El-Gohary 2013)

Semantic information element	Definition
Subject	An ontology concept representing a thing (e.g., building element) that is subject to a particular requirement
Compliance checking attribute	An ontology concept representing a specific characteristic of a "subject" that is checked for compliance
Deontic operator indicator	A term/phrase that indicates the deontic type of the requirement (i.e., obligation, permission, or prohibition)
Quantitative relation	A term/phrase that defines the type of relation for the quantity (e.g., extend)
Comparative relation	A term/phrase for comparing quantitative values, including "greater than or equal to," "greater than," "less than or equal to," "less than," and "equal to"
Quantity value	A numerical value that defines the quantity
Quantity unit	The unit of measure for a "quantity value"

Deep Learning

Deep learning methods use computational models such as deep neural networks to learn multiple levels of information representations from large-scale data (LeCun et al. 2015). Deep learning methods have drastically improved the state-of-the-art performance in many domains such as natural language processing and computer vision, and meanwhile reduced or eliminated the manual effort in feature engineering compared to traditional machine learning methods. Deep learning methods have been used in the AEC domain for solving computer vision problems such as construction equipment detection (Kim et al. 2017), activity recognition (Luo et al. 2018), and crack detection (Park et al. 2019), and text analysis problems such as building-code requirement extraction (Zhang and El-Gohary 2019). The most commonly used deep neural networks include convolutional neural networks (Kim et al. 2017; Luo et al. 2018; Gulgec et al. 2019) and recurrent neural networks (Zhang and El-Gohary 2019).

Transfer Learning

Transfer learning aims to use machine-learning models that are trained for one task and/or on the data from one domain for another task and/or on the data from another domain (Shin et al. 2016). By enabling the training of the machine learning models on large-scale, pattern-rich, and annotated training data that are outside the target domain (e.g., the AEC domain) for solving domain-specific tasks, transfer learning techniques can be used to improve the robustness and scalability of the machine learning-based methods (e.g. Teh et al. 2017) and to reduce the cost of preparing domain-specific training data. Commonly adopted transfer learning strategies adopted in the deep learning-based methods include: (1) a deep neural

network model is first trained on one training dataset, and then the trained model is fine-tuned using the other training dataset (Shin et al. 2016); (2) a deep neural network model is first trained on one training dataset, and then the first several layers of the trained model are used for extracting features for the other dataset (Lopes and Valiati 2017); (3) a deep neural network model is trained on two training datasets alternatingly (Yang et al. 2017); and (4) pretrained word embeddings are used in the input word-embedding layer of a deep neural network model.

PROPOSED MACHINE LEARNING-BASED APPROACH FOR SEMANTICALLY-ENRICHED BUILDING-CODE SENTENCE GENERATION

The proposed machine learning-based approach for generating semantically-enriched building-code sentences consists of four main steps, as shown in Figure 1.

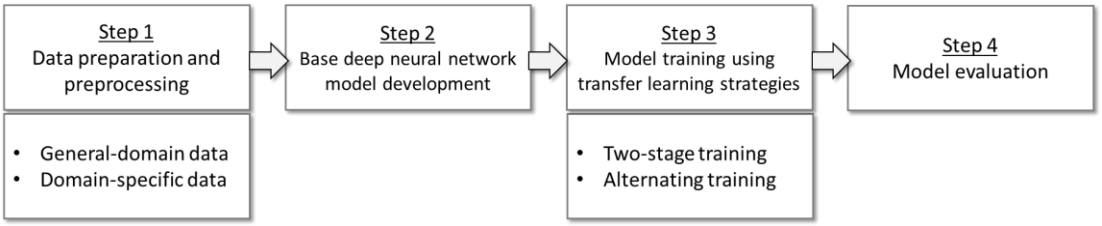
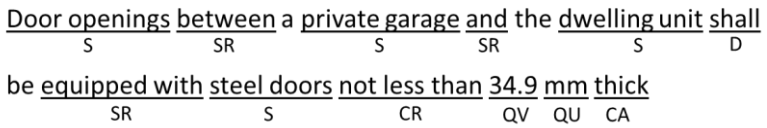


Figure 1. Proposed machine learning-based method for generating semantically-enriched building-code sentences

Step 1: Data Preparation and Preprocessing

Two types of data were collected from outside of and within the AEC domain: the general-domain data and the domain-specific data. For general-domain data, a total of 20,000 sentences from the Penn Treebank (Marcus et al. 1993) were used, which consist of sentences collected from the Wall Street Journal that are annotated with part-of-speech (POS) tags. The Penn Treebank data are large in scale, rich in syntactic and semantic patterns, and already annotated, and thus are suitable for training the deep neural networks. A POS tag indicates the syntactic role that a word plays in the sentence and can be used for semantic analysis of text data [e.g., regulatory information extraction from building code (Zhang and El-Gohary 2013)]. For domain-specific data, a total of 300 building-code sentence fragments were selected from multiple chapters of the IBC 2009 and the Champaign 2015 IBC Amendments, and were converted into semantically-enriched forms by annotating the sentences with semantic information elements (Zhang and El-Gohary 2013) (with an added semantic information element – subject relation – to describe the semantic relations between subjects) and syntactic fillers. Figure 2 shows the semantically-enriched form of an example building-code sentence. The entire general-domain dataset was used for training the deep neural network models in Step 3; and the domain-specific dataset was split into a 90:10 ratio for training and testing the deep neural network models in Steps 3 and 4, respectively.

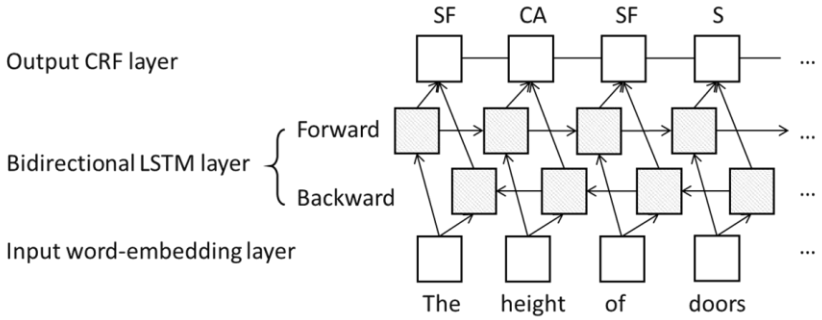


S = subject; SR = subject relation; D = deontic operator indicator; CR = comparative relation; QV = quantity value; QU = quantity unit; CA = compliance checking attribute

Figure 2. Example semantically-enriched building-code sentence

Step 2: Base Deep Neural Network Model Development

The deep neural network model – bidirectional long short term memory (LSTM) with conditional random fields (CRF) (Huang et al. 2015) – was adopted as the base model for generating semantically-enriched building-code sentences given natural language building-code sentences. The base deep neural network model consists of three main layers: the input word-embedding layer, the bidirectional LSTM layer, and the output CRF layer, as depicted in Figure 3. The input word-embedding layer aimed to represent the semantics of each word in a vector of real numbers for deep neural network computation purposes. The LSTM layer aimed to learn the feature representations of each word using the input word embeddings of the current word and the context words. To improve the ability of the LSTM layer to deal with long-term syntactic dependencies in the sentences, the bidirectional architecture was used – both the forward and backward context words were considered when learning the feature representations. Finally, for each word, the output CRF layer aimed to compute the conditional probabilities of different types of semantic annotations (i.e., the semantic information elements and the syntactic fillers), based on which the final type of semantic annotation can be predicted, given the feature representations of this word learned by the LSTM layer. To compute the model parameters, the cross entropy loss was minimized. The model was implemented using Keras in Python 3, and run on top of TensorFlow.



LSTM = long short term memory; CRF = conditional random fields; SF = syntactic filler; CA = compliance checking attribute; S = subject

Figure 3. Base deep neural network model for generating semantically-enriched building-code sentences

Step 3: Model Training Using Transfer Learning Strategies

To enable the training of the deep neural network model on both the general-domain training data and the domain-specific data, the base model was modified and

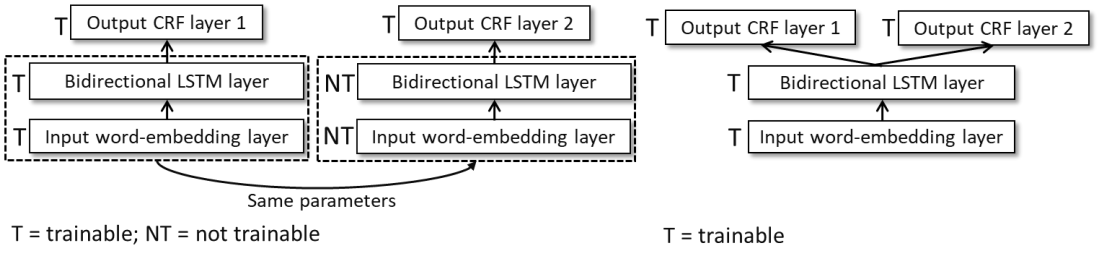
trained based on two transfer learning strategies: the two-stage training strategy and the alternating training strategy.

Two-stage Training Strategy

In the two-stage training strategy (as illustrated in Figure 4a), the deep neural network model was trained in two related stages. In the first stage, the model was trained on the general-domain data. In the second stage, the output CRF layer of the trained model (i.e., CRF 1) was replaced by a new output CRF layer (i.e., CRF 2), and the model was trained on the domain-specific data. During the training in the second stage, only the output CRF layer (CRF 2) was trainable, and the input word-embedding layer and the bidirectional LSTM layer were not trainable – the parameters of these two layers remained unchanged. For each stage, the training was stopped if the difference between the training losses of two consecutive training epochs is smaller than 0.01, or the training reaches 50 epochs.

Alternating Training Strategy

In the alternating training strategy (as illustrated in Figure 4b), the deep neural network model was trained on the general-domain and the domain-specific training data in an alternating manner. The model had two separate output CRF layers – one layer is used when the model is trained on the general-domain training data (i.e., CRF 1), and the other layer is used when the model is trained on the domain-specific training data (i.e., CRF 2). In each training iteration, there is an alternating probability p that the model was trained on a selected batch of the general-domain data, and a probability of (1-p) that the model was trained on a selected batch of the domain-specific training data. Typically, the alternating probability p is a number close to 1, meaning the model is more frequently trained on the general-domain data rather than the domain-specific data, to prevent overfitting on the relatively small-scale domain-specific data. The training was stopped if the difference between the training losses of two consecutive epochs when the model is trained on the domain-specific data is smaller than 0.01, or the training on the domain-specific data reaches 50 epochs.



(a) **(b)**

Figure 4. Deep neural network model training using two transfer learning strategies

Step 4: Model Evaluation

Given a natural language building-code sentence and a trained deep neural network model, the corresponding semantically-enriched building-code sentence was

generated by searching the optimal sequence of semantic annotations (i.e., semantic information elements and syntactic fillers) that maximizes the sum of the conditional log-likelihoods computed by the output CRF layer. The searching was conducted using dynamic programming on the matrix of conditional probabilities, allowing the optimal sequence of semantic enrichments to be generated in polynomial time instead of exponential time.

Three metrics were used to evaluate the performance of the deep neural network models for generating the semantically-enriched building-code sentences: precision, recall, and F1 measure, where for a specific type of semantic enrichment SE, TP is the number of true positives (i.e., number of words correctly labeled as SE), FP is the number of false positives (i.e., number of words incorrectly labeled as SE), and FN is the number of false negatives (i.e., number of words not labeled as SE but should have been) (Zhai and Massung 2016).

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

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

PRELIMINARY EXPERIMENTAL RESULTS

Performance of the Proposed Approach with Different Transfer Learning Strategies

The two different transfer learning strategies – the two-stage training strategy and the alternating training strategy – were tested. The deep neural network model for generating semantically-enriched building-code sentences achieved better performance when the alternating training strategy was used, outperforming the model using the two-stage training strategy by 15% in precision, 11% in recall, and 14% in F1 measure, as shown in Table 2. The deep neural network model using the two-stage training strategy achieved relatively low performance. This is possibly because the input word-embedding layer and the bidirectional LSTM layer of the deep neural network model were trained on the general-domain data only, and therefore might not have been able to learn the representations that capture the syntactic and semantic patterns in the domain-specific data well.

Table 2. Performance of the Proposed Approach with Different Transfer Learning Strategies (and Using an Alternating Probability of 92% for Alternating Training Strategy)

Transfer learning strategy	Precision[1]	Recall[1]	F1 measure[1]
Two-stage training	73%	75%	73%
Alternating training	**88%**	**86%**	**87%**

[1]Bolded font indicates the highest performance.

Performance of the Proposed Approach Using the Alternating Training Strategy with Different Alternating Probabilities

When the alternating training strategy was used for training the deep neural network model, different alternating probabilities were tested, including 90%, 92%, 95%, and 99%. The optimal performance for semantically-enriched building-code sentence generation was achieved when the alternating probability was 92%, as shown in Table 3. However, the differences were small. Further testing is needed in the future to study the statistical and practical significances of these performance differences. Comparing to a medium alternating probability (i.e., 92%), when the alternating probability was very high (i.e., 99%), the performance decreased by 13% in precision, 6% in recall, and 10% in F1 measure, possibly because the input word-embedding layer and the bidirectional LSTM layer of the deep neural network model was mainly trained on the general-domain data and might not be able to learn the representations that capture the syntactic and semantic patterns in the domain-specific data well. Comparing to a medium alternating probability (i.e., 92%), when the alternating probability was lower (i.e., 90%), the performance started to decrease slightly. This is possibly because the deep neural network model was overly trained on the domain-specific training data, and thus was overfitted to these data.

Table 3. Performance of the Proposed Approach Using the Alternating Training Strategy with Different Alternating Probabilities

Alternating probability	Precision[1]	Recall[1]	F1 measure[1]
90%	87%	85%	86%
92%	**88%**	**86%**	**87%**
95%	83%	82%	82%
99%	75%	80%	77%

[1]Bolded font indicates the highest performance.

Error Analysis

Two main types of errors were identified based on the experimental results. First, the proposed approach had errors when generating semantically-enriched forms of multiword expressions, which consist of multiple words and function as individual syntactic and semantic units, especially those including prepositions. For example, the words in the multiword expression "means of egress" should have been annotated with a single semantic information element – a subject, but the proposed model annotated the expression with a subject, a syntactic filler, and another subject. In future work, a multiword expression list could be integrated into the proposed approach for generating semantically-enriched building-code sentences. Second, the proposed approach had errors when dealing with some compliance checking attributes. For example, "Group R-1", which means the first residential group in terms of use and occupancy classification in the IBC, were usually mistakenly annotated as part of a subject instead of a compliance checking attribute. In the future, additional levels of input embedding layers could be used (e.g., character embeddings) to capture useful patterns beyond the syntactic and semantic ones.

CONCLUSION

This paper proposed a machine learning-based method for generating semantically-enriched building-code sentences for semantic analysis of the code for supporting automated compliance checking. First, a bidirectional LSTM-CRF model was adopted for the semantically-enriched building-code sentence generation task. Second, two transfer learning strategies were used for training the deep neural network models on both the general-domain data and the domain-specific data. Third, the proposed approach was tested on the domain-specific data. The proposed method achieved a precision of 88%, a recall of 86%, and an F1 measure of 87% when the second transfer learning strategy – the alternating training strategy – was used, indicating good semantically-enriched building-code sentence generation performance.

This paper contributes to the body of knowledge in three primary ways. First, the paper proposed a new, machine learning-based approach to generate semantically-enriched building-code sentences for facilitating the semantic analysis of the code for supporting automated compliance checking. Second, the paper leveraged large-scale, pattern-rich annotated training data from outside the AEC domain by using transfer learning strategies to increase the robustness and scalability of the proposed approach. Third, the experimental results show that the transfer learning strategies and some of the hyperparameters (e.g., the alternating probability for the alternating training strategy) of the deep neural network models could contribute to the performance variations of the models.

In their future work, the authors first plan to improve the proposed approach for generating semantically-enriched building-code sentences by including more types of semantic information elements, such as restrictions and references, as semantic annotations. Second, the authors will explore further ways to improve the performance of the proposed approach, including testing different general-domain training data, using more domain-specific training data, exploring different transfer learning strategies (e.g., initiating the parameters of the input word-embedding layer using pretrained word embeddings), and integrating a domain ontology into the proposed approach. Third, and most importantly, the authors plan to integrate the proposed approach for semantically-enriched building-code sentence generation with machine learning-based information extraction and semantic information matching, with an aim to find a scalable method for fully automated compliance checking.

ACKNOWLEDGEMENT

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

Abel, F., Gao, Q., Houben, G. J., and Tao, K. (2011). “Semantic enrichment of twitter posts for user profile construction on the social web.” Ext. Semant. Web, Springer, Berlin, German, 375-389.

Dugas, M., Meidt, A., Neuhaus, P., Storck, M., and Varghese, J. (2016). "ODMedit: uniform semantic annotation for data integration in medicine based on a public metadata repository." BMC Med. Res. Methodol., 16(1), 65.

Gulgec, N. S., Takáč, M., and Pakzad, S. N. (2019). "Convolutional neural network approach for robust structural damage detection and localization." J. Comput. Civil Eng., ASCE, 33(3): p.04019005.

Kim, H., Kim, H., Hong, Y. W., and Byun, H. (2017). "Detecting construction equipment using a region-based fully convolutional network and transfer learning." J. Comput. Civil Eng., ASCE, 32(2): p.04017082.

Huang, Z., Xu, W., and Yu, K. (2015). "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). "Deep learning." Nature, 521(7553), 436.

Luo, X., Li, H., Cao, D., Dai, F., Seo, J., and Lee, S. (2018). "Recognizing diverse construction activities in site images via relevance networks of construction-related objects detected by convolutional neural networks." J. Comput. Civil Eng., 32(3), p.04018012.

Lopes, U. K., and Valiati, J. F. (2017). "Pre-trained convolutional neural networks as feature extractors for tuberculosis detection." Comput. Biol. Med., 89, 135-143.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). "Building a large annotated corpus of English: The Penn Treebank." Comput. Linguist., 19(2), 313-330.

Park, S., Bang, S., Kim, H., and Kim, H. (2019). "Patch-based crack detection in black box images using convolutional neural networks." J. Comput. Civil Eng., 33(3), p.04019017.

Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R. M. (2016). "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE T. Med. Imag., 35(5), 1285-1298.

Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). "Distral: Robust multitask reinforcement learning." Adv. Neur. In., 4496-4506.

Yang, Z., Salakhutdinov, R., and Cohen, W. W. (2017). "Transfer learning for sequence tagging with hierarchical recurrent networks." arXiv preprint arXiv:1703.06345.

Zhai, C., and Massung, S. (2016), *Text data management and analysis: a practical introduction to information retrieval and text mining*, ACM, New York, USA.

Zhang, J., and El-Gohary, N. (2013). "Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking." J. Comput. Civil Eng., 30(2), p.04015014.

Zhang, R., and El-Gohary, N. (2019). "A machine learning-based approach for building code requirement hierarchy extraction." Proc., 7th CSCE Int. Constr. Spec. Conf., CSCE, Montreal, Canada.

Zhou, P., and El-Gohary, N. (2017). "Ontology-based automated information extraction from building energy conservation codes." Automat. Constr., 74, 103-117.