A Machine Learning Approach for Compliance Checking-Specific Semantic Role Labeling of Building Code Sentences

67

Ruichuan Zhang and Nora El-Gohary

Abstract

Existing automated building code checking systems/software highly rely on human interpretation of the code. Different methods have been proposed and implemented to make building codes automatically analyzable and interpretable by computers. These methods have achieved different levels of automation in rule formalization, but they all require some level of human involvement to capture the compliance checking-related semantic information in the natural language building code sentences. For example, the state-of-the art approaches require human annotation or require human effort to develop annotation and/or extraction rules. To reduce the human effort in automated code interpretation, this paper proposes a machine learning-based approach to automatically label the semantic roles in building code sentences for supporting code compliance checking. The proposed method consists of three primary elements: (1) capturing the syntactic and semantic features of the building code sentences using natural language processing techniques; (2) adapting out-of-domain training data to the task at hand based on data similarity; and (3) performing semantic role labeling using a conditional random field (CRF) model. The proposed approach was tested on a corpus of annotated text from the International Building Code, and achieved promising global precision.

Keywords

Code checking • Machine learning • Natural language processing • Semantic role labeling

67.1 Introduction

Existing automated building code checking systems/software (e.g., the Solibri Model Checker) highly rely on human interpretation of the code. An expert needs to first read and interpret the text and then formalize the requirements in the form of computer-processable rules. Such approaches, although achieve some levels of automation, are still labor-intensive and time-consuming. To address this problem, different methods have been proposed and implemented to make building codes automatically analyzable by computers. These methods have achieved different levels of automation in rule formalization, but they all require some level of human involvement to capture the semantic information in the natural language building code sentences—especially for the syntactically and semantically complex sentences. For example, the state-of-the art approaches are mostly either annotation-based or rule-based. Annotation-based approaches (e.g., [1]) require human annotation to identify and annotate the regulatory concepts that describe the building code requirements. Rule-based approaches (e.g. [2]) require human effort to develop annotation rules for automatically conducting the annotation and/or extraction rules for extracting the target information. For both approaches, eliminating the human involvement totally is challenging, because such accurate and

R. Zhang (✉) · N. El-Gohary
University of Illinois at Urbana-Champaign, 205 North Mathews Ave, Urbana, IL 61801, USA
e-mail: rzhang65@illinois.edu

N. El-Gohary
e-mail: gohary@illinois.edu

I. Mutis and T. Hartmann (eds.), *Advances in Informatics and Computing in Civil and Construction Engineering*,
https://doi.org/10.1007/978-3-030-00220-6_67

complete annotation and information extraction involves a deep level of semantic information analysis that requires complete understanding of building code requirement sentences—which are rich and complex in semantics and syntactics.

In comparison to rule-based approaches, machine-learning-based approaches aim to replace the hand-crafted-rule-based methods [3]. They aim to automatically capture the underlying patterns of the text data, by learning from a large size of text data. Supervised machine learning has been used in the domains of text mining, natural language processing, and computational linguistics to better understand the text data and extract information from such data.

To avoid/reduce the human effort in automated rule formalization for supporting automated compliance checking, a machine learning-based approach that includes three primary components is proposed: (1) semantic role labeling: automatically identifying the compositional semantics of building code sentences and automatically label the semantic roles in the sentences; (2) information extraction: automatically extracting the target semantic information based on the semantic roles, without the need for extraction rules; and (3) rule formalization: transforming the extracted information into rules. This paper focuses on semantic role labeling. A machine learning-based semantic role labeling approach is proposed, which consists of three primary elements. First, natural language processing techniques are used to capture the syntactic and semantic features of the building code sentence. Second, out-of-domain training data are adapted to the task at hand based on data similarity. Third, a conditional random field (CRF)-based algorithm is proposed and used for semantic role labeling. This paper presents the proposed approach and discusses the preliminary experimental results.

67.2 Background

67.2.1 Semantic Role Labeling

Sematic role labeling is a shallow semantic text analysis task that aims to extract the proposition units, where each unit consists of a target verb and all the constituents in the sentences that fill a semantic role of the verb [4]. The semantic roles include (1) numbered argument (A): arguments that are required or occur frequently for a verb; (2) modifiers (AM): modifiers are adverbs, adverb phrases, and prepositional phrases that describe and/or modify the verb [e.g., location (LOC), modal verb (MOD), extent (EXT), manner (MNR), and direction (DIR)]; and (3) reference (R). The numbered arguments and their grammatical definitions are summarized in Table 67.1. Semantic role labeling often acts as the cornerstone for further, much deeper annotation, understanding, and information extraction of the text. For example, the labeled semantic roles can be used for deeper IFC-oriented information annotation and/or extraction: the verbs would be candidates for IFC relationships and the noun phrases in arguments would be candidates for IFC objects and/or properties. The labeled semantic roles can also be used in entity-relationship conceptual modeling approaches such as the resource description framework (RDF), which models information in the form of subject–predicate–object which is similar to verb-argument.

67.2.2 Supervised Learning-Based Sequence Labeling

Supervised learning is one type of machine learning problems, where a function that maps input data (usually with features) to an output (such as categories) based on given input-output pairs (labeled training data) is learnt [5]. Different supervised learning algorithms can be used to learn the function from the training data. Among all the algorithms, conditional random fields (CRF) is designed specifically to deal with sequence labeling problems (e.g., semantic role labeling) [6], where given a

Table 67.1 Numbered argument

Semantic role	Grammatical definition	Example sentence from building code[a,b]
Agent (A0)	The entity that performs the action	"**The area of a membrane structure** shall not exceed the limitations in Table 503"
Patient (A1)	The entity that undergoes the action	"The area of a membrane structure shall not exceed **the limitations in Table 503**"
Verb-specific argument (A2)	Other entities that occur frequently for the action	"Draft curtains shall be constructed of **sheet metal**"

[a] Semantic role is bolded
[b] International Building Code 2009

sequence of words, a sequence of tags/roles which represent syntactic compositions (e.g., part-of-speech tags) and/or semantic compositions (e.g., semantic roles, sentiment levels, domain-specific semantic tags, etc.) needs to be found. CRF is a discriminative classifier built on the joint probability of the sequence of labels given the observed sequence of words. CRF is trained by maximum likelihood estimation, and a trained CRF assigns a probability distribution over all possible sequences of labels given a sequence of words [6]. The optimal sequence is the sequence of labels that have the maximum probability.

67.2.3 Parsing

Constituency parsing aims to organize words in a sentence into nested constituents based on phrase structures. The results of constituency parsing are often represented in the form of a tree, where the nodes are phrase structure categories, and the leaves are part-of-speech (POS) tags and words. Dependency parsing aims to analyze the grammatical structure of a sentence by linking the head words to words which modify those heads. A link indicates the grammar dependency from the current word to the head word. The results of both natural language processing techniques are usually used as features in machine learning problems to further analyze text data.

67.3 Semantic Role Labeling in Automated Compliance Checking

In the proposed approach, semantic role labeling aims to segment the whole building code sentence into several proposition units, each consisting of several semantic roles.

These semantic roles could further help in automatically extracting the target semantic information in a code sentence (e.g., subject of requirement, compliance checking attribute, etc.), without the need for extraction rules. The proposition units and the semantic role labels of the following example sentence are shown in Fig. 67.1: “at least three sides of each such larger tier shall have continuous horizontal openings not less than 30 inches in clear height extending for at least 80 percent of the length of the sides”. The example sentence is segmented as two proposition units, where Proposition Unit 2 acts as A1 (i.e., the patient) in Proposition Unit 1.

67.4 Proposed Machine Learning Approach for Semantic Role Labeling of Building Code Sentences

The proposed semantic role labeling approach consists of three primary elements: (1) natural language processing techniques are used to capture the syntactic and semantic features of the building code sentence, (2) out-of-domain training data are adapted to the task at hand based on data similarity, and (3) a CRF-based algorithm is used for semantic role labeling. The research methodology, thus, consists of five primary steps: data preparation and preprocessing, training data adaptation, feature preparation, CRF-based semantic role labeler training, and evaluation (see Fig. 67.2).

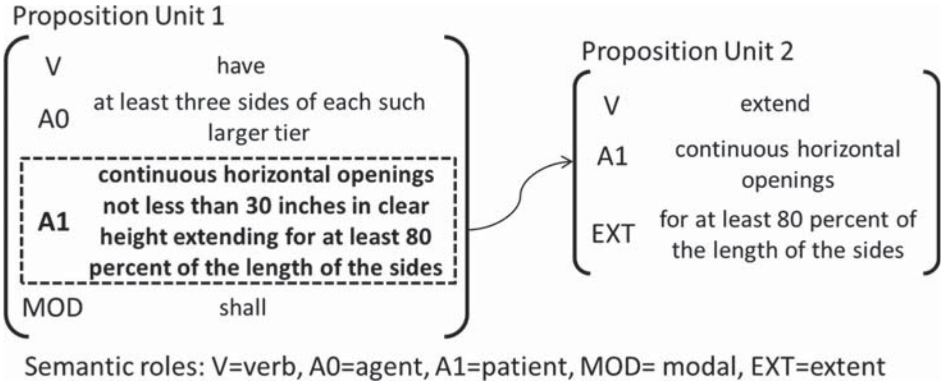
Proposition Unit 1
V
have
A0
at least three sides of each such larger tier
A1
continuous horizontal openings not less than 30 inches in clear height extending for at least 80 percent of the length of the sides
MOD
shall
Proposition Unit 2
V
extend
A1
continuous horizontal openings
EXT
for at least 80 percent of the length of the sides
Semantic roles: V=verb, A0=agent, A1=patient, MOD= modal, EXT=extent

Fig. 67.1 The proposition units and the semantic role labels in an example sentence

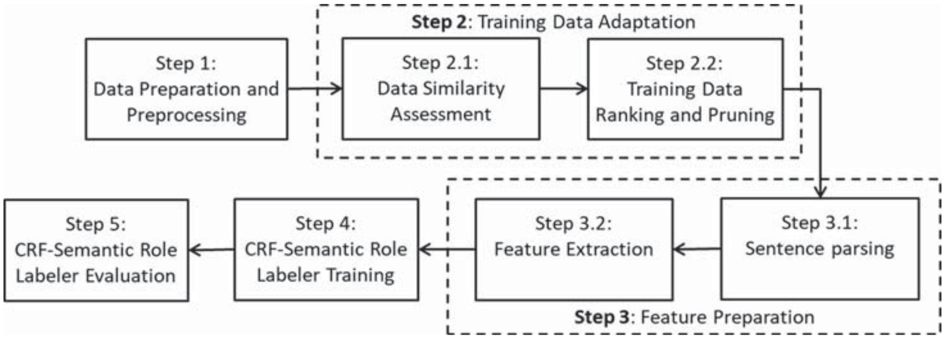
Step 2: Training Data Adaptation
Step 1: Data Preparation and Preprocessing
Step 2.1: Data Similarity Assessment
Step 2.2: Training Data Ranking and Pruning
Step 5: CRF-Semantic Role Labeler Evaluation
Step 4: CRF-Semantic Role Labeler Training
Step 3.2: Feature Extraction
Step 3.1: Sentence parsing
Step 3: Feature Preparation

Fig. 67.2 Proposed machine learning approach for semantic role labeling of building code

67.4.1 Data Preparation and Preprocessing

For out-of-domain training data, the PropBank training dataset was used [7], which contains over 30,000 sentences from the 1989 Wall Street Journal that were annotated with semantic roles.

For training data adaptation and testing, around 300 sentences were randomly selected from the International Building Code (IBC) 2009 (IBC 2009). The sentences were randomly sampled from different sentence types, in terms of syntactic and semantic structures and sentence computability [8], to better evaluate the performance and scalability of the semantic role labeler. The sentences were then manually annotated. Figure 67.1 shows an example sentence, along with its semantic role labels.

Three steps of data preprocessing were then conducted to facilitate the subsequent feature generation and training data adaptation: tokenization, lowercasing, and stemming. Tokenization aims to split the whole sentence into units (e.g., words, punctuations). Stemming aims to reduce the derived words to their root forms.

67.4.2 Training Data Adaptation

The out-of-domain training dataset was adapted to the domain-specific machine learning task at hand based on data similarity. The adaptation methodology includes two steps: data similarity assessment and data ranking and pruning.

Data Similarity Assessment. Data similarity aims to assess the similarity between the out-of-domain data (i.e., the sentences in the PropBank dataset) and the domain-specific data (i.e., the sentences in the building code). The proposed similarity between two sentences is defined as the average of one minus the normalized levenshtein distance between the POS tag sequences of the two sentences, and the cosine similarity between the two sentences. The levenshtein distance aims to capture the syntactic-level dissimilarity between two sentences. The cosine similarity aims to capture the word-level similarity between two sentences.

Data Ranking and Pruning. The out-of-domain training data were pruned based on the semantic similarity. For each training sentence, the average and maximum of the similarities to all the testing sentences were calculated. All the training sentences were then ranked based on these averages and maximums. If a training sentence ranks at the bottom 25% for both average similarity and maximum similarity, the training sentence is pruned. This step removed more than 25% of all the training sentences, and a total of 20,000 sentences, which contain over 50,000 annotated propositional units were kept and used for preparing the features and training the CRF model.

67.4.3 Feature Preparation

Parsing. All sentences were parsed using the Stanford CoreNLP constituency and dependency parser [9]. The constituency parser tags the sentences using the Penn Treebank tag set. The tag set includes: (1) word-level tags, i.e., POS tags, such as nouns (NN), and numeric values (CD), (2) phrase-level tags such as noun phrase (NP), verb phrase (VP), prepositional phrase (PP), and (3) clause-level tags such as a simple clause not introduced by subordinating conjunction or a wh-word

(S) [10]. The sentences are then represented by phrase structure rules (e.g., “S → NP VP”, which means the sentence is composed of a noun phrase followed by a verb phrase). The dependency parser tags the sentences using the Universal Dependency set. The set includes clausal argument relations such as passive nominal subject (nsubjpass), nominal modifier relations such as nominal modifier (nmod/nn), and other relations such as conjunct.

Feature Extraction. Three types of features were extracted from the original text data, and the results of both constituency and dependency parsing: plain text features, constituency-tree-derived features, and dependency-graph-derived features. Plain text features include the current, the preceding, and the following word, and also the predicate verb and the relevant distance from the predicate verb to the current word. Constituency-tree-derived features include: (1) POS tags of the current, the preceding, and the following word, and the predicate verb; (2) phrase-level tags of the current word and predicate verb, and the phrase-level tags corresponding to the lowest common parent node of these two words in the constituency parsing tree; (3) phrase structure rules of the current word and predicate verb, and the phrase structure rules corresponding to the lowest common parent node of the two words in the constituency parsing tree; and (4) the relevant height of the lowest common parent node. Dependency-graph-derived features include the head word of the current word, its POS tag, the dependency relation between the current word and the head word, and the relevant distance between the head word and the current word. Two features were further derived from each of the plain text features: the lower case of the word and the stemmed word. One other feature was further derived from each of the POS tag features: the first two letters of the original POS tag.

67.4.4 CRF Semantic Role Labeler Training

Given a sequence of words *X* in the sentence, a CRF model defines the probability of a corresponding sequence of roles *Y* as in Eq. 67.1, where the outer sum is applied over every feature function f_j, which is a function of the sentence *s*, the position *i*, the current and/or the preceding role l_i and l_{i-1}, and the inner sum is applied over every position *i* in the sentence; the length of the sentence is *n* and the total number of features is *m*; *Z* is the normalization term, and *w* are the parameters of the CRF model [6].

$$P(Y|X,w) = \frac{1}{Z}\sum_{j=1}^{m}\sum_{i=1}^{n} w_j f_j(s,i,l_i,l_{i-1}). \tag{67.1}$$

To find the optimal parameters w^* for the CRF model, given *k* training data pairs $\{(x_1, y_1),\ldots, (x_k, y_k)\}$, the conditional log likelihood of the training data with penalty is maximized, as shown in Eq. 67.2, where λ_1 and λ_2 are coefficients for L1 and L2 norm penalty applied to the parameters *w*.

$$w^* = \underset{w}{\operatorname{argmax}} \sum_{i=1}^{k} \log p(x_i|y_i,w) - \lambda_1|w| - \lambda_2\|w\|_2 \tag{67.2}$$

The whole set of training data was split into a training set and a validation set. The training set was used to train new CRF models, and the validation set was used to tune the hyper parameters λ_1 and λ_2.

67.4.5 Labeling Using CRF and Evaluation

Given a sequence of words and the trained CRF model, the labeling process aims to search the optimal sequence of roles that maximizes the sum of the conditional log likelihood. The searching is conducted using dynamic programming on the matrix of conditional probabilities. Thus, the optimal sequence of roles can be found in polynomial time instead of computing all the possible sums of conditional probabilities in exponential time.

Three metrics were used to evaluate the performance of the CRF semantic role labeler: precision (Eq. 67.3), recall (Eq. 67.4), and F1 measure (Eq. 67.5), where for a specific semantic role *L*, *TP* is the number of true positives (i.e., number of words correctly labeled as *L*), *FP* is the number of false positives (i.e., number of words incorrectly labeled as *L*), and *FN* is the number of false negatives (i.e., number of words not labeled as *L* but should have been) [11]. The evaluation metrics are computed globally, and for each type of semantic role.

$$\text{Precision } = \frac{TP}{TP+FP} \tag{67.3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{67.4}$$

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{67.5}$$

67.5 Preliminary Experimental Results and Discussion

67.5.1 Model Training

The entire training dataset was split into an 80:20 ratio, with 80% of the dataset as the training set and 20% of the dataset as the validation set. The training set was used to train the CRF model using the CRFsuite [12] built in Python. The validation set was used to tune the hyper parameters of the CRF model. Based on the validation results, 1 and 0.001 were chosen as the optimal L1 penalty and L2 penalty coefficients, respectively. To improve the computational efficiency of the training, an early-stop threshold on the number of iterations was set as 100. The precision, recall, and F1 measure of labeling the validation set are 0.72, 0.70, and 0.71, respectively.

67.5.2 Experimental Results

The trained CRF model was used to label the testing data. The global precision, recall, and F1 measure of labeling the testing data is 0.71, 0.63, and 0.65, respectively. The precision, recall, F1 measure, and the total count for each type of semantic role are shown in Table 67.2.

Table 67.2 Precision, recall, and F1 measure of each type of semantic role[a]

Semantic role	Precision	Recall	F1-measure	Count
A0	0.84	0.55	0.67	730
A1	0.70	0.78	0.74	3183
A2	0.64	0.71	0.67	1524
A3	0.23	0.73	0.34	37
ADV	0.21	0.23	0.22	227
DIR	0.20	0.62	0.30	13
EXT	0.00	0.00	0.00	34
LOC	0.69	0.19	0.30	884
MNR	0.84	0.26	0.39	560
MOD	0.95	0.93	0.94	187
NEG	0.98	0.98	0.98	49
PNC	0.25	0.28	0.26	131
TMP	0.59	0.53	0.56	188
V	0.99	0.99	0.99	485
R	1.00	0.67	0.80	36
Global/Total	0.71	0.63	0.65	8268

[a] *Semantic roles* A0 = agent, A1 = patient, A2 = verb-specific argument, A3 = other arguments, ADV = adverb, DIR = direction, EXT = extent, LOC = location, MNR = manner, MOD = modal, NEG = negation, PNC = purpose, TMP = temporal, V = verb, R = reference

Table 67.3 Confusion matrix of semantic role labeling results[a]

Semantic role	A0	A1	A2	ADV	LOC	MNR	TMP	NEG	MOD	V
A0	404	131	20	0	0	0	0	0	0	0
A1	23	2494	62	54	41	6	15	0	1	1
A2	0	261	1076	0	7	0	0	0	0	1
ADV	0	86	16	52	0	1	3	0	0	0
LOC	16	139	71	96	169	0	0	0	0	0
MNR	0	80	172	13	17	143	28	0	0	0
TMP	0	50	12	0	4	5	100	0	0	0
NEG	0	0	0	0	0	0	0	48	0	0
MOD	0	0	0	0	0	0	0	0	174	2
V	0	2	2	0	1	0	0	0	0	478

[a] *Semantic roles* A0 = agent, A1 = patient, A2 = verb-specific argument, A3 = other arguments, ADV = adverb, LOC = location, MNR = manner, TMP = temporal, NEG = negation, MOD = modal, V = verb

67.5.3 Error Analysis

The confusion matrix (Table 67.3) shows the distribution of mislabeled words in each type of semantic role. The model performs generally well when labeling numbered arguments [e.g., agent (A0), patient (A1), and verb-specific argument (A2)]. Given the fact that arguments over A2 are relatively rarely used in semantic role labeling, A0, A1, and A2 are the main focus of the analysis, and arguments with a number larger than two were neglected. The model performed relatively better in labeling A0 and A2, but sometimes confused A1 to A0 or A2. The model performed worse when labeling the modifiers [e.g., adverb (ADV), location (LOC), manner (MNR), temporal (TMP), etc.], except for the straightforward ones such as negation (NEG) and modal (MOD). The model confused ADV and LOC with A1, and confused MNR with A2, because (1) A1 and A2 can be prepositional phrases and most of the adverbs, locations, and manners are prepositional phrases; (2) the modifiers such as adverbs, locations, and manners tend to agglomerate and thus are labeled as a single semantic role; and (3) the constituency parser and dependency parser are not completely accurate in parsing the sentences, especially sentences with long and complex prepositional phrases and thus there are noises in the features derived from the constituency tree and dependency graph. Globally, the error also comes from (1) differences between the domains. For example, verbs that are rare in the training data but appear in the testing data create difficulty in labeling verb-specific arguments; and (2) longer and more complex sentences in the testing data.

67.6 Conclusion

In this paper, a machine-learning-based approach for semantic role labeling of building code requirement sentences was proposed. The proposed approach adapts out-of-domain training data to the task at hand based on a proposed data similarity measure, and performs semantic role labeling using a CRF model trained on the adapted data. Three different groups of features were proposed and used for learning: plain text features, constituency-tree-derived features, and dependency-graph-derived features. The trained CRF model achieved a global precision of 0.71, local precisions for negation and modal over 0.95, and local precisions for manner and numbered argument (e.g., A0) near 0.85. Future research effort is needed to further improve the accuracy of the proposed machine learning-based semantic role labeling approach, especially to improve the labeling performance on verb-specific arguments and modifiers such as adverb, location, and temporal.

This paper contributes to the body of knowledge in two primary ways. First, the proposed approach allows for the use of out-of-domain training data to tackle the scarcity of domain-specific training data, while adapting such out-of-domain data to a specific domain by performing data pruning. Second, the results show that the selected multi-source features are potentially effective for semantic role labeling of building code requirements. The main limitation of this approach, however, is that it is not able to directly extract compliance checking-related information, but rather segments a whole requirement into proposition units consisting of semantic roles that can be further used to extract those information.

In future work, the authors plan to pursue a number of directions to improve the performance of semantic role labeling: (1) leverage other categories of state-of-the-art machine learning algorithms; (2) extract more types of syntactic and semantic features, and test the combination of different types of features; and (3) combine the domain knowledge (e.g., domain ontology, expert-predefined rules, IFC concepts) with the proposed machine learning algorithm for semantic role labeling. Most importantly, in their future work, the authors plan to proceed with machine learning-based semantic information extraction based on the semantic roles.

References

1. Hjelseth, E., Nisbet, N.: Capturing normative constraints by use of the semantic mark-up RASE methodology. In: Proceedings of CIB W78-W102 Conference, pp. 1–10. (2011)
2. Zhang, J., El-Gohary, N.: Automated information transformation for automated regulatory compliance checking in construction. J. Comput. Civ. Eng. **29**(4), B4015001 (2015)
3. Jurafsky, D., Martin, J.: Speech and Language Processing, 3rd edn. Pearson, London (2014)
4. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 shared task: Semantic role labeling. In: Proceedings of the ninth conference on computational natural language learning, pp. 152–164. (2005)
5. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall, USA (2010)
6. Cohn, T., Blunsom, P.: Semantic role labelling with tree conditional random fields. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, pp. 169–172. Association for Computational Linguistics, USA (2005)
7. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: a corpus annotated with semantic roles. Computational Linguistics Journal **31**, 1 (2005)
8. Zhang, R., El-Gohary, N.: A clustering approach for analyzing the computability of building code requirements. In: Construction Research Congress 2018, pp. 86–95. American Society of Civil Engineers, USA (2018)
9. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55–60. Association for Computational Linguistics, USA (2014)
10. Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: annotating predicate argument structure. In: Proceedings of the workshop on Human Language Technology, PP. 114–119. (1994)
11. Zhai, C., Massung, S.: Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining. Morgan & Claypool, New York (2016)
12. Naoaki Okazaki: CRFsuite: a fast implementation of Conditional Random Fields (CRFs), http://www.chokkan.org/software/crfsuite/. Last accessed 21 Apr 2018