Named Entity Recognition Algorithm for iBISDS Using Neural Network

Ning Wang, S.M.ASCE[1]; Raja R. A. Issa, Ph.D., F.ASCE[2];
and Chimay J. Anumba, Ph.D., F.ASCE[3]

[1]Ph.D. Candidate, Rinker School of Construction Management, Univ. of Florida, Gainesville, FL. ORCID: https://orcid.org/0000-0003-3096-2385. Email: n.wang@ufl.edu
[2]Distinguished Professor, Rinker School of Construction Management, Univ. of Florida, Gainesville, FL. Email: raymond-issa@ufl.edu
[3]Dean and Professor, College of Design, Construction and Planning, Univ. of Florida, Gainesville, FL. Email: anumba@ufl.edu

ABSTRACT

Conversational Artificial Intelligence (AI) systems have become more and more popular to provide information support for human daily life. However, the construction industry still lags other industries in developing a conversational AI system to support construction activities. The developed intelligent Building Information Spoken Dialogue System (iBISDS) is a conversational AI system that provides a speech-based virtual assistant for construction personnel with considerable building information to support construction activities. The iBISDS enables construction personnel to use flexible spoken natural language queries instead of detecting exact keywords. To build an iBISDS, it is necessary to understand the intents of natural language queries for building information. This research aims to develop a named entity recognition (NER) algorithm for iBISDS to recognize and classify keywords within natural language queries. A dataset with 2,008 building information-related natural language queries was developed and manually annotated for training and testing. A Neural Network (NN) deep learning method was trained to recognize named entities within natural language queries. After training, the developed NER algorithm was applied to the testing dataset which achieved a precision of 99.74, a recall of 99.87, and an F1-score of 99.81. The preliminary result indicated that the developed NER algorithm can recognize named entities within the natural language queries accurately. This research will facilitate the further development of conversational AI systems in the construction industry.

INTRODUCTION

In Industry 4.0, Artificial Intelligence (AI) technologies facilitate the generation of conversational intelligent systems. An increasing number of companies and organizations are developing conversational AI systems to support human daily life. A Spoken Dialogue System (SDS) is a human-machine interactive conversational AI system that enables humans to converse with machines by spoken natural language (Park and Kang 2019). However, compared to other industries, the research on SDS is limited in the construction industry. An intelligent Building Information Spoken Dialogue System (iBISDS) has become necessary for construction personnel. The iBISDS developed in this study is a conversational AI system that provides building information support for construction project team members. Construction personnel can use flexible natural language speech to query information from building information models via iBISDS. To build the iBISDS, the first task is to understand queries' intents. Natural Language

Understanding (NLU) is one of the major modules in the iBISDS. The NLU module aims to detect and classify different keywords (i.e., content words) within each building information-related natural language query. The process of identification and classification of keywords is named entity recognition (NER). NER has been implemented in many disciplines including biochemistry (Akkasi and Varoglu 2017), medicine (Wang and Guan 2020), and construction regulatory areas (Zhong et al. 2020). NER has also been widely implemented in many query answering systems to recognize users' intents.

This research developed named entity recognition for iBISDS to identify and classify different keywords within natural language queries. Traditional NER methods were grammar rule-based, which required Part-of-Speech tagging (Lin et al. 2016). Traditional NER was developed based on semantic and syntactic natural language processing (NLP) analysis to analyze relationships between content words (Wu et al. 2019). However, traditional methods restricted the syntax patterns of queries, which means that traditional NER has limited scalability for queries in out-of-domain sentence patterns. With the increase of computation capability of the hardware, machine learning (ML) methods have become more and more popular for NER. To build an ML-based NER requires a large amount of training and testing data but building information-related natural language queries for iBISDS are limited in the construction industry. Therefore, this research started with the dataset development for ML-based NER. There were 2,008 building information-related natural language queries developed and annotated manually for the dataset. This research implemented Neural Network (NN) to train and test the NER for iBISDS. A Python package *SpaCy* was utilized to develop the NN. This research conducted 10 individual experiments for the 10 different random splits of the training dataset for cross-validation. After training and cross-validation, the developed NER algorithm with the best performance was finally evaluated on the testing dataset. This research will facilitate the development of iBISDS and further the adoption of conversational AI technologies in the construction industry.

LITERATURE REVIEW

Named Entity Recognition. Named entity recognition (NER) is a task of natural language processing (NLP), which aims to recognize predefined entities from unstructured texts, such as a person, organization, and location (Du and Zhao 2020). Predefined entities are also called named entities which define text annotation categories. Named entities are the fundamental elements for further analysis (Akkasi and Varoglu 2017). For example, NER was implemented to recognize and extract disease name, duration, and drug name from doctor-patient question answering communications in the medical area (Wang and Guan 2020). Recognized entities are utilized to analyze related information from natural language texts. There are two categories of NER methods in the construction area: grammar rule-based NER and machine learning (ML)-based NER. Grammar rule-based NER methods are based on NLP Part-of-Speech (POS) tagging to analyze semantic and syntactic information within unstructured texts. Lin et al. (2016) developed a query-answering method based on NLP syntax grammar trees to extract named entities for analysis of natural language queries. Zhou and El-Gohary (2017) proposed a semantic and syntactic NLP method to extract information from energy conservation codes for compliance checking. Wu et al. (2019) also utilized the POS tagging NLP method to extract different types of keywords (i.e., named entities) to analyze natural language queries for information retrieval from the BIM object database. Grammar rule-based NER methods are based on traditional NLP methods like syntax rules and ontologies. However, building such rules and ontologies is complex and time-consuming

(Du and Zhao 2020). Grammar rule-based NER methods are highly dependent on a grammar tree, which is not always exclusive for each natural language sentence. Grammar rule-based NER methods are limited for sentences with more than one grammar tree representation.

In recent years, due to the increase of hardware computation capabilities, ML-based NER has become more and more popular. ML-based NER methods are based on the Neural Network (NN) training instead of traditional NLP semantic and syntactic rules. Compared to grammar rule-based NER, ML-based NER is not dependent on grammar trees but based on a large amount of labeled training data. ML-based NER enables a machine to recognize named entities after NN training. ML-based NER have been implemented in many construction areas. Liu and El-Gohary (2017) proposed a NER method of semi-supervised learning with conditional random fields to extract information, like bridge element and maintenance material, from bridge inspection reports. Zhong et al. (2020) proposed a NER method to recognize predefined procedural constraints from construction regulations. Moon et al. (2020) proposed a NER model to extract bridge damage information, such as bridge element, damage, and cause, from inspection reports. Baker et al. (2020) developed a NER method to extract information about accident precursors from construction injury reports. ML-based NER was also implemented to recognize named entities from construction specifications for automated specifications review (Moon et al. 2021).

BIO Scheme. Existing ML-based NER methods have trained NNs to identify and classify words (i.e., named entities) within natural language documents in the construction area. However, most existing research does not follow any annotation scheme. Each word within a sentence is considered as one entity, but in most cases, one entity may contain more than one word. For example, "building information modeling" should be recognized as one entity instead of three separate entities, although there are three words within one entity. For existing ML-based NER methods, each word within "building information modeling" was considered as, for example, a "concept", which means "building" is a "concept". However, a NER algorithm also needs to recognize entity boundaries (i.e., word chunks). For such a situation, annotation schemes become necessary, which define word chunks within a sentence. There are many NER annotation schemes, such as Beginning-Inside-Outside (BIO) (Trivedi and Majhi 2020). The BIO annotation scheme was developed by Ramshaw and Marcus (1995). "B" represents the beginning word of a named entity, "I" stands for a word inside a named entity, and "O" represents a word outside the named entity. For the example of "building information modeling", "building" should be recognized as "B-concept", and "information" and "modeling" should be recognized as "I-concept". For a human, "building" and "information" belong to one entity, while for a machine, "building" and "information" are considered as different entities. With the annotation scheme, "building" and "information" are two entities when training a NN, which increases the accuracy of NER. Otherwise, a machine may treat "building" and "information" as the same entity without an annotation scheme. Due to its simple tagging mechanism, the BIO scheme is one of the most popular annotation schemes (Riaz et al. 2020). This research implemented the BIO scheme for the NER development.

EXPERIMENT

Hardware and Software Environment. The hardware environment used to train the NN was a CPU Intel Core i7-10700 of base 2.90 GHz with 8 cores and 32.0 GB 2666 MHz RAM. The integrated development environment (IDE) was Jupyter Notebook 6.2.0, and the Python interpreter was Python 3.8.5 distributed by Anaconda. Jupyter Notebook was used to preprocess training and testing data. The NN was trained in Command Line Interface (CLI), and the performances of developed NNs were compared using the same datasets in CLI. This research utilized GPU acceleration for training and testing the NN to increase the efficiency of computation. The GPU was an Nvidia GeForce RTX 2060 super 8 GB GDDR6 with 2176 CUDA Cores. The NER algorithm was developed based on the open-source Python package *spaCy*. *SpaCy* is a natural language processing library which a transformer-based training system for text classification, Part-of-Speech tagging, and named entity recognition training (spaCy 2021).

Training and Testing Data Generation. The training and testing datasets for natural language understanding of building information-related queries are limited in the construction industry. To train a NN for named entity recognition of iBISDS, this research developed a dataset with 2,008 building information-related queries for training and testing. The dataset was developed for information retrieval from building information models for iBISDS, so the developed queries are information retrieval-related ones which were not covering model manipulation queries or others. Each query in the dataset was developed based on the IFC4 ADD2 TC1 schema and regarding one academic building Rinker model and one sample building information model in IFC format. For example, in the IFC4 schema, an *IfcDoor* contains attributes, such as *overallheight*, *objecttype*, and *overallwidth*. One query can be generated based on the schema, like "What is the height of the door in faculty office 0316?", and "faculty office 0316" is the location information for the door. The developed dataset includes two types of queries: queries for attribute information (e.g., the height of the door) and queries for quantities information. Queries for quantities information are regarding the quantities of building elements, such as doors and windows. For example, "How many doors of single flush C are on 03 fl 03 top of slab?" is a query regarding quantities information. Since there are so many building elements in building information models, this research focused on doors, windows, rooms, and building stories. The developed dataset contains 839 queries for attribute information and 1,169 queries for quantities information about building elements. The sentence structures (i.e., syntax) of queries in the developed dataset are flexible. Queries for quantities information could be "How many doors in faculty office 0316", "What are the quantities of doors …", or "Is there a door …". Queries for grammar rule-based NER methods are pattern-based ones which require users to utilize exact sentence structure with limited option. The developed dataset for ML-based NER contains more flexible sentence structures, which means users can utilize more flexible natural language queries.

The developed dataset needs to be manually annotated for the training and testing of the NER NN. To annotate each query in the dataset, this research developed six annotation labels: TYPE, ENTITY, ATTRIBUTE, OBJECT, LOC-ROOM, and LOC-LEVEL (see Table 1). The six NER annotation labels were developed based on the system requirements of iBISDS. The label TYPE is to annotate the type of the building element within one query, such as window, door, and building story, since the iBISDS requires the TYPE entity to locate the corresponding IFC type (e.g., *IfcWindow*). The label ENTITY is to annotate the exact building element, like level 2, since the iBISDS requires such ENTITY information to locate the queried IFC entity. The label ATTRIBUTE is to annotate the attribute information, such as height and width, which can be used to locate the target IFC attribute value of the queried IFC entity. The label OBJECT is to annotate object type information, for example, a door of "single flush B". The recognized "single flush B" is an object type of a door, and the phrase was annotated as OBJECT. The recognized OBJECT can be used to locate the target object type of the queried IFC type. The labels LOC-ROOM and LOC-LEVEL were utilized to annotate the location information of building elements, like "classroom 0112" or "level 3". The iBISDS utilized the location information to find the correspon-

-ding IFC type and entity. This research used Doccano for manual annotation of the dataset. Doccano is an open-source text annotation tool for text classification, named entity recognition, and sequence to sequence natural language processing tasks (Nakayama et al. 2018). The manual annotation of the developed dataset also followed the BIO annotation scheme, which means one named entity can contain more than one word (see Figure 1). The start-end spans of named entities within each query were manually annotated in Doccano. The developed dataset with annotations was in the JSON format which was published on GitHub (https://github.com/wangningstar/iBISDS/tree/main/BINLQ).

Table 1. NER Annotation Labels Categories

NER Labels	Descriptions	Examples
ENTITY	Building element entity phrase	top of parapet, FL 01, 02 top of slab, 321556, 03 top of slab
TYPE	Building element type phrase	floor, level, window, door, story, storey, building storey
ATTRIBUTE	Building element attribute phrase	elevation, height, width, height, object type, long name
OBJECT	Object type phrase	fixed Rinker, louver 12, single flush B, double flush, casement
LOC-ROOM	Room location information	main office, classroom 0112, faculty office 0331, room 1, shop
LOC-LEVEL	Level / Building story location information	level 3, floor 03, top of parapet

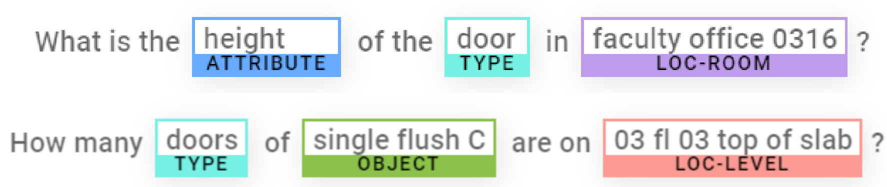

Figure 1. Examples of Annotations in the Doccano Interface

Data Preprocessing. The developed dataset was randomly split into training and testing datasets. The training dataset is 70% of the entire dataset (i.e., 1,405 queries), while the testing dataset is 30% of the dataset (i.e., 603 queries). Table 2 shows experimental data statistics about the numbers of queries and named entities within training and testing datasets. The training dataset was further randomly split into *train* and *validation* datasets. The *train* dataset is for the NN training purpose, while the *validation* dataset is to avoid the overfitting problem of a machine learning algorithm. The *validation* dataset can provide an unbiased evaluation of a trained algorithm and adjust parameters in the NN to avoid overfitting to the *train* data (Bishop 2006). The *validation* dataset is 30% of the training dataset (i.e., 422 queries), while the *train* dataset is 70% of the training dataset (i.e., 983 queries). The *train* and *validation* datasets were randomly split from the training dataset 10 times for cross-validation. Cross-validation was used to compare performances of different machine learning models and pick the one with the best performance (Murphy 2012). This research conducted 10 individual experiments using 10 different combinations of the *train* and *validation* dataset to pick the NN with the best performance for final testing use. The testing dataset (i.e., 603 queries) remained the same during the cross-validation and was utilized for the final evaluation of the NN with the best performance after the 10 experiments. This research implemented the Python package *Sklearn* to randomly split the developed dataset into three datasets. In addition, since *spaCy* was utilized to train the NN, *train*, *validation*, and testing datasets required to be converted into ".spacy" format from JSON format to fulfill the data requirements of *spaCy*. This research implemented the *DocBin* function from *spaCy* for the conversion of those datasets.

Table 2. Experimental Data Statistics

Dataset	Queries	ENTITY	TYPE	ATTRIBUTE	OBJECT	LOC-ROOM	LOC-LEVEL
Training	1,405	338	1,100	582	741	470	384
Testing	603	143	467	257	332	190	156

Training and Validating. One-time NN training is not reliable, so this research conducted 10 experiments with 10 different combinations of *train* and *validation* datasets which were randomly split from the training dataset. This research implemented *spaCy* to train and validate the NN. *SpaCy* supports the Command Line Interface (CLI) training process based on gradient descent and backpropagation NN methods. The setting of parameters of NNs was stored in a configuration file (i.e., .cfg file), such as training epoch, evaluation frequency, and dropout rate. The configuration file is an open-source file that can be easily accessed and modified. This research set training parameters: epochs as 30, evaluation frequency as 20, dropout rate as 0.1, and optimizer as "Adam" in the configuration file. The selected NER NN model was "spacy.TransitionBasedParser.v2" which contains 64 hidden NN layers. The corpus for word embedding is "spacy.Corpus.v1" in this research. The word embedding process is to convert the tokenization of one train sample into a vector for a machine to process. The selected method for tokenization to vector was "spacy-transformers.TransformerListener.v1". based on the "RoBERTa-base". The other parameter settings were using *spaCy* default ones. During each experiment, the 10 combinations of *train* and *validation* datasets were utilized for the *spaCy* CLI training process. This research utilized a CUDA GPU to accelerate the training process.

Table 3. Cross-Validation Statistics (Best Performance)

Experiment #	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
Precision (%)	99.91	100.00	99.82	99.63	99.63	**100.00**	99.81	99.91	99.72	99.91
Recall (%)	100.00	99.91	99.82	99.63	99.82	**100.00**	100.00	99.91	99.82	99.91
F1 score (%)	99.95	99.95	99.82	99.63	99.73	**100.00**	99.90	99.91	99.77	99.91

All trained NNs were saved, and the performances were recorded during the experiments. This research saved two trained NN models during each experiment: the best model and the last model. During the 30 epochs of the training process, each training NN model has the best stage and last stage. The best model means the trained NN can provide the best performance during each experiment. The performances of NNs were measured by precision, recall, and F1 score. F1 score is the harmonic mean of precision and recall (Murphy 2012). Also, the F1 score is a widely utilized index to measure the performance of a trained NN model or algorithm (Moon et al. 2021). This research implemented the F1 score to measure and compare the 10 trained NNs. Table 3 shows the cross-validation statistics of each trained NN with the best performance in each experiment. To pick the best NN from the 10 experiments, this research selected the trained NN with the highest F1 score for the final evaluation on the testing dataset.

RESULTS AND DISCUSSION

After the 10 individual experiments of training and cross-validation, the trained NN from experiment #E6 achieved the best performance (i.e., highest F1 score) than other NNs. Therefore, this research used the NN from experiment #E6 for the final evaluation by the testing dataset. Jupyter notebook was used to visualize the named entity recognition of one query from the testing dataset. Figure 2 shows the prediction of the trained algorithm on one example testing query. In the example, the trained NN recognized "window" as TYPE label, "louver 12" as OBJECT label, and "medium classroom 0103" as LOC-ROOM label. This research also implemented *spaCy* to evaluate the trained NN #E6 via the CLI evaluation function. The testing dataset was converted into the ".spaCy" format to be input into the evaluation function. The tested NN was the best model from experiment #E6. Also, this research utilized a CUDA GPU to accelerate the testing process. After the testing, the trained NN gained a precision of 99.74, a recall of 99.87, and an F1-score of 99.81. Each annotation category also had a testing result of precision, recall, and F1-score (see Table 4). The label "LOC-LEVEL" achieved 100.00 for precision, recall, and F1-score. There were misclassifications for other labels. For example, the phrase "faculty office" can be correctly classified in the LOC-ROOM label in a query of "How many windows in the faculty office 0321?". However, in a query of "How many faculty offices are on the third floor?", the same phrase "faculty office" was misclassified into the same LOC-ROOM label instead of the correct label OBJECT. The reason for misclassification was that the same phrase (e.g., faculty office) should be classified into two labels (e.g., LOC-ROOM and OBJECT) in the two different queries, but the NN misclassified them into the same label.

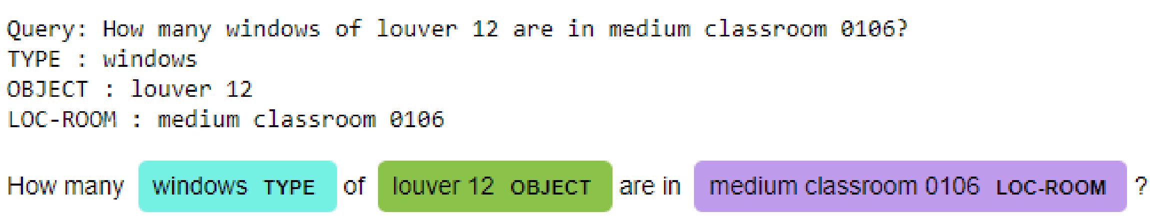

Figure 2. An Example of NER Result in Jupyter Notebook

Table 4. NER Statistics (Best Performance)

Categories	Precision (%)	Recall (%)	F1 score (%)
ENTITY	99.30	99.30	99.30
TYPE	99.79	100.00	99.89
ATTRIBUTE	99.61	100.00	99.81
OBJECT	100.00	99.70	99.85
LOC-ROOM	99.48	100.00	99.74
LOC-LEVEL	100.00	100.00	100.00

CONCLUSIONS

The development of conversational AI technologies has led to the growing popularity of spoken dialogue systems in everyday life. However, the construction industry still lags other industries in developing a spoken dialogue system for construction project team members. An intelligent iBISDS becomes necessary to provide information support for construction personnel. IBISDS enables construction personnel to query quantity and attribute information about building elements from building information models by flexible spoken natural language. Named entity recognition (NER) is one of the major functions of iBISDS in understanding the intents of natural language queries. This research developed an ML-based NER instead of the traditional NER on grammar rules which restricted users' input. A dataset was developed for training and testing the ML-based NER algorithm in this research. There were 2,008 building information-related queries generated and annotated manually. The building information queries were manually developed based on the IFC schema and annotated based on predefined labels. This research developed six predefined annotation labels based on the system requirements of iBISDS. The manual annotation was also following the BIO annotation scheme. The developed dataset includes queries for attribute information and queries for quantity information about building elements like windows, doors, rooms, and building stories. This research randomly split the dataset into training and testing datasets. The training dataset has been further randomly split into *train* and *validation* datasets for cross-validation.

This research conducted 10 experiments with 10 different random combinations of *train* and *validation* datasets. *SpaCy* Neural Network algorithm was implemented to train and evaluate trained NER models. After the training and cross-validation, the NER algorithm with the best performance from experiment #E6 was selected for the final evaluation on the testing dataset. After the testing, the trained NER algorithm achieved a precision of 99.74, a recall of 99.87, and an F1-score of 99.81. The experimental result indicated the developed NER NN can accurately recognize and classify different keywords in building information-related queries with flexible sentence structures. The developed NER method also has limitations. Since the dataset with annotations was specifically developed for iBISDS, the F1-score is much higher than other related NER research. The developed dataset only focused on the attribute and quantity information of building components. More types of building information-related queries need to be generated and collected to develop a more diverse dataset which is one of the future research directions. Also, the dataset contains two types of queries: attribute and quantity related. To better understand the intents of natural language queries from users, future research will develop an ML-based text classification method based on the developed dataset by recurrent neural networks. The developed NER NN is the basis of the natural language understanding of iBISDS. This research will facilitate the development of conversational AI systems and further the adoption of machine learning methods in the construction industry.

REFERENCES

Akkasi, A., and Varoglu, E. (2017). "Improving Biochemical Named Entity Recognition Using PSO Classifier Selection and Bayesian Combination Methods." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Institute of Electrical and Electronics Engineers Inc., 14(6), 1327–1338.

Baker, H., Hallowell, M. R., and Tixier, A. J. P. (2020). "Automatically learning construction injury precursors from text." *Automation in Construction*, Elsevier B.V., 118, 103145.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
Du, Y., and Zhao, W. (2020). "Named Entity Recognition Method with Word Position." *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*, 154–159.
Lin, J., Hu, Z., Zhang, J., and Yu, F. (2016). "A Natural-Language-Based Approach to Intelligent Data Retrieval and Representation for Cloud BIM." *Computer-Aided Civil & Infrastructure Engineering*, 31(1), 18–33.
Liu, K., and El-Gohary, N. (2017). "Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports." *Automation in Construction*, Elsevier B.V., 81, 313–327.
Moon, S., Chung, S., and Chi, S. (2020). "Bridge Damage Recognition from Inspection Reports Using NER Based on Recurrent Neural Network with Active Learning." *Journal of Performance of Constructed Facilities*, 34(6), 04020119.
Moon, S., Lee, G., Chi, S., and Oh, H. (2021). "Automated Construction Specification Review with Named Entity Recognition Using Natural Language Processing." *Journal of Construction Engineering and Management*, 147(1), 04020147.
Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., and Liang, X. (2018). "{doccano}: Text Annotation Tool for Human." <https://github.com/doccano/doccano>.
Park, Y., and Kang, S. (2019). "Natural Language Generation Using Dependency Tree Decoding for Spoken Dialog Systems." *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., 7, 7250–7258.
Ramshaw, L. A., and Marcus, M. P. (1995). *Text Chunking using Transformation-Based Learning*.
Riaz, F., Anwar, M. W., and Muqades, H. (2020). "Maximum Entropy based Urdu Named Entity Recognition." *2020 International Conference on Engineering and Emerging Technologies, ICEET 2020*, Institute of Electrical and Electronics Engineers Inc.
spaCy. (2021). "Industrial-Strength Natural Language Processing." <https://spacy.io/>(Apr. 6, 2021).
Trivedi, I., and Majhi, S. (2020). "Span level model for the construction of scientific knowledge graph." *Proceedings of the 2020 International Conference on Computing, Communication and Security, ICCCS 2020*, Institute of Electrical and Electronics Engineers Inc.
Wang, Z., and Guan, H. (2020). "Research on Named Entity Recognition of Doctor-Patient Question Answering Community Based on BiLSTM-CRF Model." *Proceedings - 2020 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020*, Institute of Electrical and Electronics Engineers Inc., 1641–1644.
Wu, S., Shen, Q., Deng, Y., and Cheng, J. (2019). "Natural-language-based intelligent retrieval engine for BIM object database." *Computers in Industry*, Elsevier, 108, 73–88.
Zhong, B., Xing, X., Luo, H., Zhou, Q., Li, H., Rose, T., and Fang, W. (2020). "Deep learning-based extraction of construction procedural constraints from construction regulations." *Advanced Engineering Informatics*, Elsevier Ltd, 43.
Zhou, P., and El-Gohary, N. (2017). "Ontology-based automated information extraction from building energy conservation codes." *Automation in Construction*, Elsevier B.V., 74, 103–117.